Теория надёжности обучения по прецедентам
Курс лекций

К. В. Воронцов
http://www.MachineLearning.ru - Участник:Vokov
voron@forecsys.ru

Материал находится в стадии разработки, может содержать ошибки и неточности. Автор будет благодарен за любые замечания и предложения, направленные по адресу vokov@forecsys.ru, либо высказанные в обсуждении страницы «Теория надёжности обучения по прецедентам (курс лекций, К.В.Воронцов)» вики-ресурса www.MachineLearning.ru.

Содержание

1 Проблема переобучения и слабая вероятностная аксиоматика

Машинное обучение или *обучение по прецедентам* — это наука о том, как научить компьютер решать задачи прогнозирования и принятия решений в условиях, когда знаний о предметной области не хватает для построения обоснованных математических моделей, но зато имеются значительные массивы эмпирических данных.

Типичный пример — задачи медицинской диагностики. Здесь под данными понимаются электронные истории болезни. Данные об отдельном клиническом случае включают анамнез, результаты обследований, назначенные лечебные мероприятия, показатели результативности лечения, и т. д. Требуется построить алгоритм, который на основе имеющейся информации о новом пациенте мог бы поставить диагноз, рекомендовать лечение или предсказать исход заболевания. Задача *обучения по прецедентам* заключается в том, чтобы построить такой алгоритм на основе *обучающей выборки* — совокупности прецедентов, наблюдавшихся в прошлом, для которых правильные диагнозы (решения, исходы) уже известны.

Другой пример — задача *кредитного скоринга*. Здесь прецеденты — это заявки на получение кредита в банке. Анкетные данные заявителя включают: возраст, пол, образование, профессию, стаж работы, доход семьи, размер задолженностей в других банках, наличие телефона, и т. д. Задача состоит в том, чтобы по обучающей выборке заёмщиков с уже известной кредитной историй построить алгоритм, предсказывающий, будут ли у данного заявителя проблемы с погашением кредита.

Нетрудно построить алгоритм, который выдаёт правильные решения для объектов обучающей выборки. Например, он мог бы сравнивать новый объект с каждым из обучающих объектов, и в случае полного совпадения выдавать правильное решение, записанное в таблице исходных данных, а во всех остальных случаях выбирать случайное решение. Очевидно, такой алгоритм никого не устроит. Для успешного обучения важно не только запоминать, но и обобщать. Способность алгоритмов правильно находить общие закономерности по частным эмпирическим данным называют *обобщающей способностью* (generalization ability).

Основная задача *теории статистического обучения*[1] (statistical learning theory, SLT) заключается в том, чтобы давать статистически обоснованные количественные оценки обобщающей способности и затем на их основе конструировать обучаемые алгоритмы, надёжно работающие вне материала обучения.

Одна из основных проблем SLT — относительно низкая точность оценок. Чрезмерно осторожные оценки, рассчитанные на худший случай, на практике могут приводить к ошибочным решениям. В данном курсе лекций рассматривается новое направление в SLT — комбинаторная теория обобщающей способности, позволяющая более точно оценивать вероятность переобучения.

---

[1]Второе название — *теория вычислительного обучения* (computational learning theory, COLT). Различия между COLT и SLT незначительны и довольно условны. В частности, COLT включает в себя проблематику вычислительной сложности алгоритмов обучения.

§1.1 Обучение и переобучение

Пусть задано конечное множество объектов $\mathbb{X} = \{x_1, \ldots, x_L\}$, называемое *генеральной выборкой*; множество A, элементы которого называются *алгоритмами*, и бинарная функция $I\colon A \times \mathbb{X} \to \{0, 1\}$, называемая *индикатором ошибки*. Если $I(a, x) = 1$, то говорят, что алгоритм a ошибается на объекте x.

Число ошибок алгоритма a на выборке $X \subseteq \mathbb{X}$ есть

$$n(a, X) = \sum_{x \in X} I(a, x).$$

Частотой ошибок алгоритма a на выборке $X \subseteq \mathbb{X}$ называется отношение

$$\nu(a, X) = \frac{n(a, X)}{|X|}.$$

Задача обучения по прецедентам. Допустим, что генеральная выборка разбита на две подвыборки, $\mathbb{X} = X \sqcup \bar{X}$. Выборка X называется *наблюдаемой* или *обучающей*, для объектов $x \in X$ известны значения индикатора ошибки $I(a, x)$. Выборка $\bar{X}$ называется *скрытой* или *контрольной*, и на ней значения индикатора ошибки неизвестны. Задача состоит в том, чтобы найти алгоритм $a \in A$ с минимальным числом ошибок на генеральной выборке $n(a, \mathbb{X})$, пользуясь только информацией о наблюдаемой выборке. Данная задача в общем случае не имеет точного решения, поскольку алгоритм a выбирается по неполной информации. Поэтому ставится задача поиска приближённого решения и оценивания его точности.

Методом обучения[2] называется отображение $\mu\colon 2^{\mathbb{X}} \to A$, которое произвольной обучающей выборке $X \subset \mathbb{X}$ ставит в соответствие некоторый алгоритм $a = \mu X$ из A.

Наиболее естественная стратегия обучения — найти алгоритм, допускающий наименьшее число ошибок на обучающей выборке. Обозначим через $A(X)$ подмножество алгоритмов a, на которых число ошибок $n(a, X)$ минимально:

$$A(X) = \operatorname{Arg}\min_{a \in A} n(a, X) = \big\{a \in A\colon n(a, X) \leqslant n(a', X),\ \forall a' \in A\big\}. \tag{1.1}$$

Если $\mu X \in A(X)$ для любого $X \subset \mathbb{X}$, то метод μ называется методом *минимизации эмпирического риска*.

Переобученностью метода μ при разбиении $X \sqcup \bar{X} = \mathbb{X}$ называется разность частот ошибок алгоритма μX на контроле и на обучении:

$$\delta_\mu(X) = \nu(\mu X, \bar{X}) - \nu(\mu X, X).$$

Если $\delta_\mu(X) \geqslant \varepsilon$ при некотором достаточно малом $\varepsilon \in (0, 1)$, то говорят, что метод μ *переобучается* на выборке X.

---

[2] В англоязычной литературе метод обучения принято называть *алгоритмом обучения* (learning algorithm) [40], а алгоритм $a\colon \mathbb{X} \to \mathbb{Y}$ — классификатором (classifier), гипотезой (hypothesis), решающей функцией (decision function), либо просто функцией (function). Термин «алгоритм» как отображение из множества объектов во множество ответов употребляется в работах научной школы академика Ю. И. Журавлёва [18]. Термины «метод» и «алгоритм», обозначающие процедуру построения функции a по выборке данных употребляются в отечественной литературе вперемежку [9, 1, 20, 19].

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	
x_1	1	1	1	**0**	0	1	1	1	
x_2	0	0	0	**0**	1	1	0	0	
x_3	0	1	1	**0**	0	0	0	0	X — наблюдаемая обучающая выборка
x_4	1	0	1	**0**	1	0	1	0	
x_5	0	0	1	**0**	1	1	0	0	
x_6	0	0	0	**1**	1	1	0	0	
x_7	1	0	0	**1**	1	1	0	0	
x_8	0	0	0	**1**	0	0	0	1	$\bar{X}$ — скрытая контрольная выборка
x_9	0	1	1	**1**	1	1	0	0	
x_{10}	0	1	1	**1**	1	1	0	0	

Рис. 1.1. Пример матрицы ошибок, $L = 10$, $D = 8$, $\ell = k = 5$. Показано одно из C_{10}^5 разбиений выборки на наблюдаемую и скрытую подвыборки. Метод минимизации эмпирического риска выбирает алгоритм a_4 и является переобученным относительно данной пары выборок, причём при любом $\varepsilon \in (0, 1)$.

Матрица ошибок. Бинарный вектор-столбец $\vec{a} = \big(I(a, x_i)\big)_{i=1}^{L}$ будем называть *вектором ошибок* алгоритма a. Совокупность всех попарно различных векторов ошибок, порождаемых алгоритмами $a \in A$, образует *матрицу ошибок* размера $L \times D$. Строки этой матрицы соответствуют объектам, столбцы — алгоритмам. Число столбцов D может быть меньше $|A|$, так как различные алгоритмы могут порождать одинаковые векторы ошибок. Множество алгоритмов A вполне может быть и бесконечным, однако число D попарно различных векторов ошибок всегда конечно и не превышает 2^L. В дальнейшем именно матрица ошибок, а не исходное множество алгоритмов A, будет для нас основным объектом исследования. Обычно будет сразу предполагаться, что A — это конечное множество алгоритмов с попарно различными векторами ошибок.

Пример 1.1. Матрица ошибок на рис. 1.1 разбита на обучающую и контрольную выборки так, что алгоритм a_4, минимизирующий эмпирический риск, допускает ошибки на всех объектах контрольной выборки. Это и есть переобучение.

Можно предположить, что в данном примере переобучение оказалось следствием неудачного разбиения выборки на обучение и контроль. В дальнейшем мы введём *вероятность переобучения* — величину, которая характеризует выборку $\mathbb{X}$ и метод обучения μ, и не зависит от случайного разбиения $X, \bar{X}$.

Задачи классификации и восстановления регрессии. Допустим, что каждому объекту $x \in \mathbb{X}$ соответствует *правильный ответ* $y(x) \in \mathbb{Y}$. Функция $y\colon \mathbb{X} \to \mathbb{Y}$ называется *целевой зависимостью* (target function).

В качестве *алгоритмов* будем рассматривать функции того же вида $a\colon \mathbb{X} \to \mathbb{Y}$, допускающие эффективную реализацию на компьютере.

В качестве множества A чаще всего задаётся некоторое параметрическое *семейство алгоритмов* $A = \{\varphi(x, \theta)\colon \theta \in \Theta\}$, где $\varphi\colon \mathbb{X} \times \Theta \to \mathbb{Y}$ — фиксированная функция, Θ — множество допустимых значений параметра θ, называемое *пространством параметров* или *пространством поиска* (search space).

В качестве *индикатора ошибки* возьмём бинарную функцию $I\colon A \times \mathbb{X} \to \{0, 1\}$, равную 1, когда предсказание $a(x)$ существенно отличается от правильного отве-

та $y(x)$. Индикатор ошибки может определяться по-разному в зависимости от постановки задачи, главным образом — от природы множества допустимых ответов $\mathbb{Y}$.

В задачах *классификации* множество классов $\mathbb{Y}$ конечно, и индикатор ошибки чаще всего задаётся в виде[3]

$$I(a,x) = \big[a(x) \neq y(x)\big], \quad x \in \mathbb{X},\ a \in A.$$

В задачах *восстановления регрессии* и многих задачах *прогнозирования* $\mathbb{Y} = \mathbb{R}$, и величину ошибки принято характеризовать непрерывной *функцией потерь* (loss function), например, квадратичной: $\mathscr{L}(a,x) = \big(a(x) - y(x)\big)^2$. Тем не менее, можно определять и бинарные функции потерь, например,

$$I(a,x) = \big[|a(x) - y(x)| \geqslant \delta(x)\big], \quad x \in \mathbb{X},\ a \in A,$$

где $\delta(x)$ — пороговый уровень, ниже которого отклонение не считается ошибкой. Заметим, что бинарная функция потерь является *робастной*, то есть нечувствительной к *выбросам* — большим отклонениям ответа алгоритма $a(x)$ от истинного $y(x)$.

Пример 1.2. Пусть объектами являются n-мерные числовые векторы, $\mathbb{X} \subset \mathbb{R}^n$. Обозначим через $\langle \xi, \theta \rangle = \xi_1\theta_1 + \cdots + \xi_n\theta_n$ скалярное произведение векторов в $\mathbb{R}^n$. *Линейные семейства алгоритмов* определяются следующим образом:

$A = \big\{a(x) = \operatorname{sign}\langle x, \theta\rangle : \theta \in \mathbb{R}^n\big\}$ — для задач классификации, $\mathbb{Y} = \{-1, +1\}$;
$A = \big\{a(x) = \langle x, \theta\rangle : \theta \in \mathbb{R}^n\big\}$ — для задач восстановления регрессии, $\mathbb{Y} = \mathbb{R}$.

Линейный алгоритм классификации представляет собой гиперплоскость с направляющим вектором θ, разделяющую пространство на две области — классы -1 и $+1$.

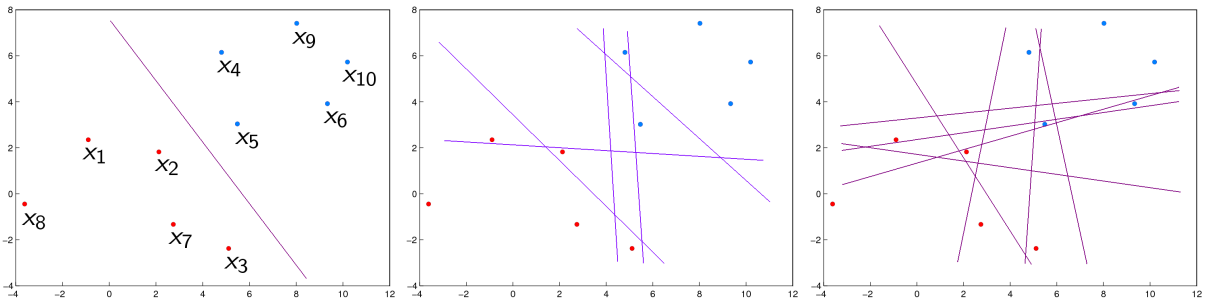
В теории переобучения понятие «алгоритма» можно не конкретизировать, предполагая лишь, что это элементы некоторого абстрактного множества A. Главное, чтобы для любого алгоритма a была возможность определить, допускает ли он ошибку на объекте x. Такое понимание «алгоритма» с одной стороны расширяет класс рассматриваемых задач, но с другой стороны ограничивает его теми задачами, в которых важен лишь факт ошибки, но не важна величина отклонения $|a(x) - y(x)|$.

Пример 1.3. На рис. 1.2 приведён пример задачи классификации с двумя классами, $|\mathbb{Y}| = 2$. Объектами являются $L = 10$ точек плоскости, по 5 объектов в каждом классе, алгоритмами — всевозможные разделяющие прямые, то есть в данном случае A — это семейство *линейных классификаторов*. В матрице ошибок содержится один нулевой столбец (объекты разделяются прямой без ошибок), 5 столбцов с одной ошибкой, 8 столбцов с двумя ошибками, и т. д.

При решении прикладных задач переобучение наблюдается практически всегда. Величина переобученности может оказаться как приемлемо малой, так и неприемлемо большой. Для управления процессом обучения хотелось бы иметь точные количественные оценки переобученности.

---

[3] Квадратные скобки переводят логическое значение в числовое: [ложь] = 0, [истина] = 1. Это очень практичное обозначение, называемое *нотацией Айверсона* [14].



x_1	0	1	0	0	0	0	1	0	0	0	0	1	1	0	...
x_2	0	0	1	0	0	0	1	1	0	0	0	0	0	0	...
x_3	0	0	0	1	0	0	0	1	1	0	0	0	0	1	...
x_4	0	0	0	0	1	0	0	0	1	1	0	0	0	0	...
x_5	0	0	0	0	0	1	0	0	0	1	1	1	0	0	...
x_6	0	0	0	0	0	0	0	0	0	0	1	0	1	0	...
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	1	...
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...
x_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...

Рис. 1.2. Пример матрицы ошибок, порождаемой семейством линейных алгоритмов классификации на выборке длины $L = 10$, содержащей 5 объектов одного класса и 5 второго. На трёх графиках вверху показаны все алгоритмы с попарно различными векторами ошибок, с числом ошибок, соответственно, 0, 1, 2.

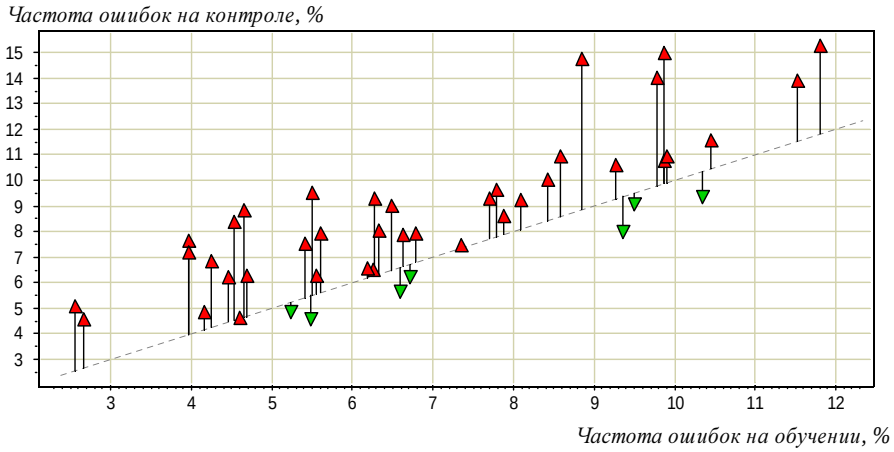


Рис. 1.3. Зависимость $\nu(\mu X, \bar{X})$ от $\nu(\mu X, X)$ для задачи прогнозирования отдалённого результата хирургического лечения атеросклероза.

Пример 1.4. На рис. 1.3 точкам соответствуют алгоритмы, построенные различными методами обучения по одной и той же выборке в задаче прогнозирования отдалённого результата хирургического лечения атеросклероза. Объектами x_i являются описания больных до проведения операции (данные гемодинамики и иммунологического обследования); ответы $y_i \in \{0, 1\}$ кодируют результат лечения через год после операции: 0 — успех, 1 — рестеноз шунта и повторная операция. По горизонтальной оси отложена частота ошибок на обучении, по вертикальной — на контроле. Наблюдается систематическое смещение точек графика вверх; почти все точки лежат выше биссектирисы. Это и есть переобучение.

Основная задача статистического обучения заключается в том, чтобы по наблюдаемой обучающей выборке найти в A алгоритм, который допускал бы как можно меньше ошибок на скрытой контрольный выборке. Вопрос можно ставить и так: какими свойствами должны обладать множество алгоритмов A и метод обучения μ, чтобы вероятность переобучения была минимальной? Чтобы получить ответ, необходимо сначала уточнить, в каком смысле здесь понимается «вероятность».

§1.2 Слабая вероятностная аксиоматика

Будем полагать, что объекты из конечного неслучайного множества $\mathbb{X}$ появляются в случайном порядке, или, другими словами, что все $L!$ перестановок генеральной выборки $\mathbb{X}$ равновероятны. Это предположение мы будем называть *слабой (или перестановочной) вероятностной аксиоматикой*. Другие вероятностные предположения нам не понадобятся. Далее мы увидим, что одного этого уже вполне достаточно для получения многих фундаментальных фактов теории вероятностей, математической статистики, теории статистического обучения.

Обозначим через S_L группу всех перестановок L элементов. Всевозможные перестановки генеральной выборки будем обозначать через $\tau\mathbb{X}$, $\tau \in S_L$.

Определение 1.1. *Пусть задан предикат $\psi\colon \mathbb{X}^L \to \{0,1\}$. Если $\psi(\tau\mathbb{X}) = 1$, то будем говорить, что событие ψ произошло на перестановке $\tau\mathbb{X}$. Вероятностью события ψ называется доля перестановок $\tau\mathbb{X}$, на которых оно произошло:*

$$\mathsf{P}_\tau\, \psi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \psi(\tau\mathbb{X}). \tag{1.2}$$

В слабой аксиоматике вероятность события зависит от состава объектов генеральной выборки $\mathbb{X}$, но не зависит от порядка их перечисления. Функция распределения и математическое ожидание также зависят от выборки.

Определение 1.2. *Пусть $\xi\colon \mathbb{X}^L \to \mathbb{R}$ — вещественная функция. Функцией распределения величины ξ на выборке $\mathbb{X}$ будем называть функцию $F_\xi\colon \mathbb{R} \to [0,1]$ вида*

$$F_\xi(z) = \mathsf{P}_\tau\big[\xi(\tau\mathbb{X}) \leqslant z\big]. \tag{1.3}$$

Определение 1.3. *Математическим ожиданием величины $\xi\colon \mathbb{X}^L \to \mathbb{R}$ на выборке $\mathbb{X}$ будем называть её среднее арифметическое по всем перестановкам τ:*

$$\mathsf{E}_\tau \xi(\tau\mathbb{X}) = \frac{1}{L!} \sum_{\tau \in S_L} \xi(\tau\mathbb{X}). \tag{1.4}$$

Заметим, что знаки P_τ и E_τ можно понимать как операцию среднего арифметического по всем перестановкам: $\mathsf{P}_\tau \equiv \mathsf{E}_\tau \equiv \frac{1}{L!}\sum\limits_{\tau \in S_L}$, то есть в слабой аксиоматике вероятность и математическое ожидание формально определяются одинаково.

Если предикат ψ является функцией двух подвыборок X, $\bar{X}$ и значение $\psi(\mathbb{X}) = \varphi(X, \bar{X})$ не зависит от порядка элементов в подвыборках X и $\bar{X}$, то *вероятность события* ψ определяется как доля разбиений генеральной выборки:

$$\mathsf{P}\,\varphi(X, \bar{X}) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \varphi(X, \bar{X}),$$

где $[\mathbb{X}]^\ell$ — множество всех ℓ-элементных подмножеств генеральной выборки $\mathbb{X}$.

Поскольку генеральная выборка $\mathbb{X}$ фиксирована и $\bar{X} = \mathbb{X} \setminus X$, в дальнейшем наряду с $\varphi(X, \bar{X})$ будем пользоваться сокращённой записью $\varphi(X)$.

Вероятность переобучения определяется как доля разбиений выборки, при которых переобученность $\delta_\mu(X)$ превышает заданный порог $\varepsilon \in (0, 1)$:

$$Q_\varepsilon \equiv Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\big[\nu(\mu X, \bar{X}) - \nu(\mu X, X) \geqslant \varepsilon\big]. \tag{1.5}$$

Вероятность переобучения характеризует обобщающую способность метода μ на выборке $\mathbb{X}$. Её оценивание будет основной задачей на протяжении всего курса.

Наряду с Q_ε будем также оценивать *вероятность большой частоты ошибок* на скрытой контрольной выборке:

$$R_\varepsilon \equiv R_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\nu(\mu X, \bar{X}) \geqslant \varepsilon\big], \tag{1.6}$$

а также среднюю (по всем разбиениям) частоту ошибок на контрольной выборке, называемую *полным скользящим контролем* (complete cross-validation, CCV) [49]:

$$C \equiv C(\mu, \mathbb{X}) = \mathsf{E}\nu(\mu X, \bar{X}). \tag{1.7}$$

Заметим, что CCV является математическим ожиданием, а $(1 - R_\varepsilon)$ — функцией распределения случайной величины $\nu(\mu X, \bar{X})$.

Обращение оценок. Пусть $Q_\varepsilon \leqslant \eta(\varepsilon)$ — верхняя оценка вероятности переобучения и существует функция $\varepsilon(\eta)$, обратная к $\eta(\varepsilon)$. Тогда $\eta(\varepsilon(\alpha)) = \alpha$ для любого $\alpha \in [0, 1]$. Следовательно, $\mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon(\eta)\big] \leqslant \eta$, и наша верхняя оценка может быть переформулирована в эквивалентном виде: с вероятностью не менее $(1-\eta)$ справедлива верхняя оценка частоты ошибок на скрытой выборке:

$$\nu(\mu X, \bar{X}) \leqslant \nu(\mu X, X) + \varepsilon(\eta). \tag{1.8}$$

Параметр ε называют *точностью*, а η — *надёжностью* оценки [6].

Аналогично, если найдена верхняя оценка $R_\varepsilon \leqslant \eta(\varepsilon)$ и $\varepsilon(\eta)$ — функция, обратная к $\eta(\varepsilon)$, то с вероятностью не менее $(1 - \eta)$ справедлива верхняя оценка

$$\nu(\mu X, \bar{X}) \leqslant \varepsilon(\eta).$$

Итак, от верхних оценок Q_ε или R_ε легко переходить к верхним оценкам частоты ошибок на контроле. Обычно из оценки (1.8) выводят новый критерий обучения, отличающийся от минимизации эмпирического риска $\nu(a, X)$ дополнительным слагаемым $\varepsilon(\eta)$. Этот приём является очень важным в SLT, так как он позволяет превращать теоретические оценки в новые оптимизационные методы обучения.

Эмпирические оценки вероятности переобучения и скользящий контроль. Напомним, что вероятность события φ мы определяем как среднее значение $\varphi(X)$ по множеству всех C_L^ℓ разбиений $\mathbb{X} = X \sqcup \bar{X}$:

$$\mathsf{P}\,\varphi(X) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \varphi(X).$$

Непосредственное вычисление этой величины возможно только при небольших значениях ℓ или k. В типичных случаях число разбиений C_L^ℓ огромно. Приближённой оценкой для $\mathsf{P}\,\varphi(X)$ является среднее по некоторому подмножеству N выборок длины ℓ, не слишком большому, чтобы сумма вычислялась за приемлемое время, но и не слишком маленькому, чтобы приближение было достаточно точным:

$$\hat{\mathsf{P}}\varphi(X) = \frac{1}{|N|} \sum_{X \in N} \varphi(X).$$

Далее символами $\hat{\mathsf{P}}$ и $\hat{\mathsf{E}}$ будет обозначаться операция усреднения по некоторому подмножеству разбиений N.

Если подмножество выборок N формируется путём случайного выбора, то говорят об оценке *методом Монте-Карло*.

В машинном обучении оценивание по подмножеству разбиений используют для эмпирического измерения обобщающей способности и называют *скользящим контролем* (cross-validation, CV). Он незаменим в тех случаях, когда теоретические оценки обобщающей способности не известны или недостаточно точны. Скользящий контроль де-факто является стандартной методикой тестирования и сравнения алгоритмов машинного обучения.

Связь с сильной вероятностной аксиоматикой. Классическая теоретико-мерная аксиоматика А. Н. Колмогорова (будем называть её сильной) основана на понятии вероятностного пространства $\langle \mathscr{X}, \Omega, \mathsf{P} \rangle$, где $\mathscr{X}$ — множество допустимых объектов, Ω — аддитивная σ-алгебра событий на $\mathscr{X}$, P — вероятностная мера, определённая на событиях из Ω. В задачах статистического анализа данных обычно предполагается, что множество объектов $\mathbb{X} = \{x_1, \ldots, x_L\}$ является *простой выборкой*, то есть объекты выбираются из множества $\mathscr{X}$ случайно и независимо согласно вероятностной мере P. Независимость означает, что вероятностная мера на множестве выборок $\mathscr{X}^L$ инвариантна относительно перестановок элементов выборки. В приложениях множество $\mathscr{X}$, как правило, бесконечно, а мера P неизвестна.

В слабой аксиоматике множество $\mathscr{X}$ всех гипотетически возможных объектов не вводится. Рассматривается только конечное множество объектов — *генеральная выборка* $\mathbb{X}$. Оно может включать в себя как объекты, наблюдавшиеся ранее, так и скрытые объекты, которые станут известны в будущем. Вероятностным пространством является конечное множество всех перестановок генеральной выборки $\mathbb{X}$, на котором задаётся равномерное распределение. Таким образом, случайными полагаются не сами объекты, а лишь порядок их появления, что соответствует предположению о независимости объектов выборки в сильной аксиоматике.

Следующая теорема утверждает, что для перевода оценки из слабой аксиоматики в сильную достаточно взять её математическое ожидание по выборке $\mathbb{X}$.

Теорема 1.1. *Пусть в слабой аксиоматике найдено значение вероятности*

$$\mathsf{P}_\tau\,\psi(\tau\mathbb{X}) = f(\mathbb{X}). \tag{1.9}$$

Тогда в сильной аксиоматике выполняется равенство

$$\mathsf{P}_{\mathbb{X}}\,\psi(\mathbb{X}) = \mathsf{E}_{\mathbb{X}} f(\mathbb{X}). \tag{1.10}$$

Доказательство. В силу независимости наблюдений в выборке $\mathbb{X}$ для произвольной перестановки τ справедливо равенство $\mathsf{P}_{\mathbb{X}}\,\psi(\mathbb{X}) = \mathsf{P}_{\mathbb{X}}\,\psi(\tau\mathbb{X})$. Возьмём среднее по всем перестановкам τ от левой и правой частей этого равенства:

$$\mathsf{P}_{\mathbb{X}}\,\psi(\mathbb{X}) = \frac{1}{L!}\sum_{\tau\in S_L}\mathsf{P}_{\mathbb{X}}\,\psi(\tau\mathbb{X}) = \mathsf{P}_\tau\,\mathsf{E}_{\mathbb{X}}\psi(\tau\mathbb{X}) = \mathsf{E}_{\mathbb{X}}\,\mathsf{P}_\tau\,\psi(\tau\mathbb{X}) = \mathsf{E}_{\mathbb{X}} f(\mathbb{X}),$$

что и требовалось доказать. ■

В случаях, когда оценка $f(\mathbb{X})$ не зависит от выборки $\mathbb{X}$, конечный результат — оценка в правой части (1.9) и (1.10) — будет одинаков в обеих аксиоматиках.

Финитарные и инфинитарные вероятности. Рассмотрим фундаментальную задачу теории вероятностей, тесно связанную с *законом больших чисел*: оценить вероятность большого отклонения частоты $\nu(S,X)$ события S на конечной выборке X от вероятности $P(S)$ данного события:

$$P_\varepsilon^S = \mathsf{P}_X\big\{|\nu(S,X) - P(S)| > \varepsilon\big\}. \tag{1.11}$$

В практических задачах анализа данных вероятность события $P(S)$ невозможно узнать точно, поскольку вероятностная мера на множестве объектов, как правило, неизвестна. Провести бесконечное число наблюдений также невозможно. В результате оказывается, что вероятность большого отклонения P_ε^S непосредственно не может быть измерена в эксперименте как частота события $\big\{X\colon |\nu(S,X) - P(S)| > \varepsilon\big\}$, поскольку само наступление этого события не может быть точно идентифицировано.

Данная проблема не возникает, если с самого начала отказаться от употребления вероятности $P(S)$. Она определяется как предел частоты $\nu(S,X')$ события S на произвольной случайной выборке X' при $|X'|\to\infty$. В то же время, практический интерес представляет именно частота $\nu(S,X')$, как величина, непосредственно наблюдаемая в эксперименте. Изменим постановку задачи (1.11) и будем оценивать вероятность большого отклонения частот события S в двух различных выборках:

$$Q_\varepsilon^S = \mathsf{P}_{X,\bar{X}}\big\{|\nu(S,X) - \nu(S,X')| > \varepsilon\big\}. \tag{1.12}$$

Если предполагать, что выборки X и X' независимы, то для определения вероятности Q_ε^S уже не нужно ни бесконечного числа испытаний, ни существования

вероятностной меры на исходном пространстве событий. Вероятность Q_ε^S является *финитарной* и может быть вычислена комбинаторными методами как доля разбиений $\mathbb{X} = X \sqcup \bar{X}$, при которых имеет место большое отклонение частот. Кроме того, она может быть измерена в эксперименте по подмножеству разбиений, так как идентификация события $\bigl\{X, X' \colon |\nu(S, X) - \nu(S, X')| > \varepsilon\bigr\}$ не вызывает затруднений.

Таким образом, вероятности $P(S)$ и P_ε^S в (1.11) имеют различную природу. Вероятность $P(S)$ принципиально *инфинитарна* — для её определения требуется либо знать вероятностную меру P на бесконечном множестве $\mathscr{X}$, либо осуществить предельный переход $\nu(S, X') \to P(S)$ при $|X'| \to \infty$, что невозможно при практическом анализе данных. Вероятность P_ε^S также инфинитарна, но после замены $P(S)$ на частоту $\nu(S, X')$ она принимает финитарный вид Q_ε^S, допускающий и точное вычисление, и эмпирическое измерение.

Эти соображения как раз и приводят к слабой вероятностной аксиоматике, запрещающей использование инфинитарных вероятностей и «событий», которые не могут быть идентифицированы в эксперименте. Приведём несколько менее формальных соображений в пользу слабой аксиоматики.

Современная теория вероятностей возникла из стремления объединить в рамках единого формализма частотное понятие вероятности, берущее начало от азартных игр, и континуальное, идущее от геометрических задач типа задачи Бюффона о вероятности попадания иглы в паркетную щель. В аксиоматике Колмогорова континуальное понятие берётся за основу как более общее. Ради этой общности в теорию вероятностей привносятся гипотезы сигма-аддитивности и измеримости — технические предположения из теории меры, имеющие довольно слабые эмпирические обоснования [2]. Таким образом, для изучения дискретных явлений, связанных со случайностью, определение вероятности как континуальной меры изначально избыточно.

Обратим внимание на замечание А. Н. Колмогорова в [22, стр. 252]: «представляется важной задача освобождения всюду, где это возможно, от излишних вероятностных допущений. На независимой ценности чисто комбинаторного подхода к теории информации я неоднократно настаивал в своих лекциях». Один из вариантов комбинаторно-алгебраического построения теории информации можно найти в книге В. Д. Гоппы [13]. Высказывание Колмогорова в значительной степени относится и к математической статистике, которая также имеет дело с конечными выборками.

Ученик А. Н. Колмогорова Ю. К. Беляев в предисловии к книге «Вероятностные методы выборочного контроля» пишет: «возникло глубокое убеждение, что в теории выборочных методов можно получить содержательные аналоги большинства основных утверждений теории вероятностей и математической статистики, которые к настоящему времени найдены в предположении взаимной независимости результатов измерений» [3, стр. 9].

Уместно привести ещё одно высказывание А. Н. Колмогорова: «Чистая математика благополучно развивается как по преимуществу наука о бесконечном... Весьма вероятно, что с развитием современной вычислительной техники будет понято, что в очень многих случаях разумно изучение реальных явлений вести, избегая про-

межуточный этап их стилизации в духе представлений математики бесконечного и непрерывного, переходя прямо к дискретным моделям» [22, стр. 239].

Асимптотические оценки. Бесконечно длинные выборки не реализуются на практике, просто потому, что конечна память компьютеров и время, отпущенное исследователям на эксперименты. В классической вероятностной аксиоматике данное обстоятельство не принимается во внимание, в частности, когда пишут

$$P(S) = \lim_{|X|\to\infty} \nu(S, X),$$

где $P(S)$ — вероятность события S, $\nu(S, X)$ — частота события S в выборке X.

В слабой аксиоматике запись $|X| \to \infty$ запрещена, и понятие вероятности события $P(S)$ не определено. Мы не вправе предполагать, что выборка реальных объектов может быть сколь угодно длинной. Тем не менее, было бы нелепо отказываться от преимуществ богатого математического аппарата асимптотического анализа.

Простой компромисс заключается в том, чтобы разрешить асимптотический анализ получаемых *численных оценок*, рассматривая его лишь как способ приближённых вычислений. Например, получив в слабой аксиоматике оценку, зависящую от длины выборки, $\mathsf{P}_\tau\,\psi(\tau\mathbb{X}) = f(L)$, мы можем исследовать асимптотическое поведение числовой функции $f(L)$ при $L \to \infty$. Очевидно, при этом нет необходимости предполагать существование сколь угодно длинной выборки.

О природе переобучения. Неформально, переобучение — это чрезмерно точная подгонка алгоритма a под конкретную обучающую выборку X в ущерб его *обобщающей способности*. Ожидается, что метод μ обнаружит некие общие закономерности генеральной выборки $\mathbb{X}$, но он находит лишь частные закономерности, характерные только для X. Переобучение носит фундаментальный характер и связано с неполнотой информации в момент применения метода μ.

Ещё одну интерпретацию переобучения даёт следующий мысленный эксперимент. Пусть задано конечное множество из D алгоритмов, которые допускают на генеральной выборке $\mathbb{X}$ одно и то же число ошибок m, независимо друг от друга. Число ошибок любого из этих алгоритмов на обучающей выборке X подчиняется одному и тому же распределению (гипергеометрическому, как будет показано далее). Выбирая алгоритм с минимальным числом ошибок s на обучающей выборке, мы фактически находим минимум из D независимых одинаково распределённых случайных величин. Математическое ожидание минимума уменьшается с ростом числа D. Следовательно, переобученность $\delta = \frac{m-s}{k} - \frac{s}{\ell} = \frac{m}{k} - s\frac{L}{\ell k}$ увеличивается с ростом D. Эти рассуждения остаются в силе и в общем случае, когда алгоритмы не являются независимыми (имеются схожие алгоритмы) и допускают различное число ошибок (имеется расслоение множества алгоритмов по уровням числа ошибок m). Оценивать эффекты *сходства* и *расслоения* довольно трудно, но именно этим мы и займёмся в дальнейшем, так как они существенно понижают вероятность переобучения.

Резюме

Введены основные понятия, которые будут использоваться на протяжении всего курса: генеральная выборка, наблюдаемая обучающая выборка, скрытая контрольная выборка, метод обучения, переобученность, матрица ошибок.

Введена слабая вероятностная аксиоматика, основанная на предположении, что все перестановки конечной генеральной выборки имеют равные шансы реализоваться. По сути, это элементарная теория вероятностей XVII-го века, которая сводится к комбинаторике. Далее мы убедимся, что многие фундаментальные статистические факты могут быть адекватно выражены в терминах частот и не нуждаются в теоретико-мерной вероятности. Слабая аксиоматика адекватна многим задачам анализа данных, поскольку она имеет дело исключительно с конечными выборками и величинами, непосредственно измеримыми в конечном эксперименте.

Поставлена основная задача — получение как можно более точных, эффективно вычислимых оценок вероятности переобучения (1.5), либо вероятности большой частоты ошибок на контроле (1.6), либо полного скользящего контроля (1.7).

В следующей лекции мы дадим точное решение основной задачи для простейшего, но важного частного случая, когда семейство A состоит из единственного алгоритма. Фактически, это означает, что никакого обучения нет. Для заданного фиксированного алгоритма по известной частоте ошибок на наблюдаемой выборке будет оцениваться частота его ошибок на скрытой выборке.

Упражнения

Задача 1.1 (1). В задаче классификации с двумя классами $\mathbb{Y} = \{-1, 1\}$ выборка задана точками прямой: $\mathbb{X} = \left\{-\frac{L}{2}, \ldots, -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \ldots, \frac{L}{2}\right\}$. Для семейства алгоритмов классификации $a_\theta(x) = \operatorname{sign}(x - \theta)$ с параметром $\theta \in \mathbb{R}$ построить матрицу ошибок и график зависимости $n(a_\theta, \mathbb{X})$ от θ, если целевая зависимость имеет вид:

1) $y(x) = \operatorname{sign} x$;
2) $y(x) = \operatorname{sign} \sin \pi x$;
3) $y(x) = \begin{cases} \operatorname{sign} \sin \pi x, & |x| \leqslant M; \\ \operatorname{sign} x, & |x| > M; \end{cases}$

Сколько различных векторов ошибок порождает данное семейство алгоритмов? Сколько из них допускают m ошибок ($m = 0, \ldots, L$) на генеральной выборке $\mathbb{X}$?

Задача 1.2 (2). В задаче классификации с двумя классами $\mathbb{Y} = \{-1, 1\}$ выборка задана точками на окружности: $\mathbb{X} = \left\{(\sin \varphi_i, \cos \varphi_i) \colon \varphi_i = \left(i - \frac{1}{2}\right)\frac{2\pi}{L},\ i = 1, \ldots, L\right\}$. Целевая зависимость имеет вид $y(x) = \operatorname{sign} \xi_2$, $x = (\xi_1, \xi_2) \in \mathbb{R}^2$. Для семейства линейных алгоритмов классификации $a_\theta(x) = \operatorname{sign}(\theta_1 \xi_1 + \theta_2 \xi_2 + \theta_0)$, с параметром $\theta = (\theta_1, \theta_2, \theta_3) \in \mathbb{R}^3$ построить (описать) матрицу ошибок. Сколько различных векторов ошибок порождает данное семейство алгоритмов? Сколько из них допускают ровно m ошибок ($m = 0, \ldots, L$) на генеральной выборке $\mathbb{X}$?

Задача 1.3 (5). Каково максимальное число различных векторов ошибок, порождаемых линейными алгоритмами классификации на выборке $\mathbb{X} = \{x_1, \ldots, x_L\} \subset \mathbb{R}^n$?

2 Оценивание частоты события и гипергеометрическое распределение

Начнём с самого простого частного случая, когда множество алгоритмов состоит из единственного элемента, $A = \{a\}$. Тогда вероятность переобучения переходит в вероятность большого отклонения частот в двух выборках. Она тесно связана с законом больших чисел, имеющим фундаментальное значение для теории вероятностей. Поэтому мы забудем ненадолго про алгоритмы и перейдём к более общей терминологии, заодно немного упростив обозначения.

§2.1 Задача оценивания (предсказания) частоты события

Пусть $S \subseteq \mathbb{X}$ — некоторое множество объектов; будем называть его «событием». Событие S и вектор ошибок алгоритма a взаимно однозначно соответствуют друг другу: $S = \{x_i \colon I(a, x_i) = 1\}$, $I(a, x_i) = [x_i \in S]$.

Обозначим через $n(U) = |S \cap U|$ *число элементов* события S на произвольной конечной выборке $U \subseteq \mathbb{X}$, а через $\nu(U) = n(U)/|U|$ — *частоту события* S на U.

Задача предсказания частоты события состоит в том, чтобы оценить частоту события S на скрытой выборке $\bar{X}$ по его частоте на наблюдаемой выборке X и оценить надёжность предсказания, то есть получить оценку вида

$$\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] \leqslant \eta(\varepsilon); \tag{2.1}$$

В тех случаях, когда S интерпретируется как «нежелательное событие», может ставиться задача получения односторонней верхней оценки:

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] \leqslant \eta(\varepsilon). \tag{2.2}$$

Лемма 2.1. *Если $n(\mathbb{X}) = m$, то число элементов события S в наблюдаемой подвыборке $n(X)$ и в скрытой подвыборке $n(\bar{X})$ подчиняются гипергеометрическому распределению:*

$$\mathsf{P}\big[n(X) = s\big] = \mathsf{P}\big[n(\bar{X}) = m - s\big] = h_L^{\ell,\,m}(s) = \frac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}, \tag{2.3}$$

где s принимает значения от $s_0 = \max\{0, m - k\}$ до $s_1 = \min\{\ell, m\}$.

Доказательство. Отобрать s элементов события S в наблюдаемую подвыборку можно C_m^s различными способами. Для каждого из этих способов имеется $C_{L-m}^{\ell-s}$ способов сформировать оставшуюся часть наблюдаемой подвыборки из объектов, не принадлежащих S. Значит, $C_m^s C_{L-m}^{\ell-s}$ — число разбиений, при которых s элементов множества S попадают в наблюдаемую подвыборку, остальные $(m-s)$ — в скрытую. Их доля в общем числе разбиений C_L^ℓ как раз и составляет $h_L^{\ell,\,m}(s)$. ■

Замечание 2.1. Если условие $0 \leqslant s \leqslant m$ не выполняется, то будем полагать, что $C_m^s = 0$. Аналогично, если не выполняется условие $0 \leqslant \ell - s \leqslant L - m$, то $C_{L-m}^{\ell-s} = 0$. Отсюда следует, что если не выполняется условие $s_0 \leqslant s \leqslant s_1$, то $h_L^{\ell,\,m}(s) = 0$.

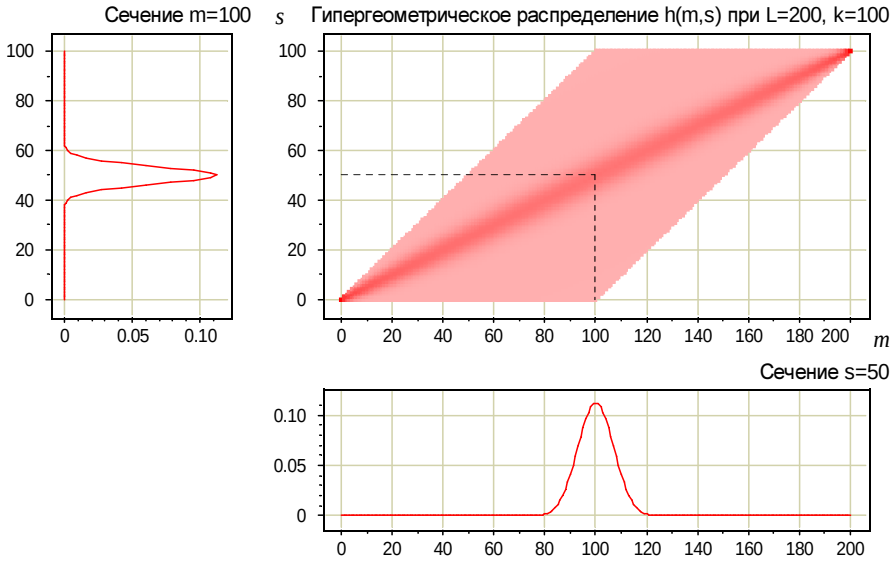


Рис. 2.1. Область определения гипергеометрической функции $h(m,s) = h_L^{\ell,\,m}(s)$ при $L = 200$, $\ell = k = 100$, $m = 30$.

§2.2 Гипергеометрическое распределение

Гипергеометрическое распределение носит фундаментальный характер и возникает во многих задачах. В данном параграфе перечисляются в справочном порядке основные свойства гипергеометрического распределения [3, 4].

1. При фиксированных L и ℓ функция $h_L^{\ell,\,m}(s) = \dfrac{C_m^s C_{L-m}^{\ell-s}}{C_L^\ell}$ определена на множестве пар целых чисел (m,s): $0 \leqslant m \leqslant L$, $\max\{0, m-k\} = s_0 \leqslant s \leqslant s_1 = \min\{\ell, m\}$. Это множество имеет форму параллелограмма, рис. 2.1. Вне этой области принято полагать $h_L^{\ell,\,m}(s) = 0$.

2. Введём следующие обозначения для сумм крайних левых и крайних правых членов гипергеометрического распределения:

$$H_L^{\ell,\,m}(z) = \sum_{s=s_0}^{\lfloor z \rfloor} h_L^{\ell,\,m}(s)\,; \qquad \bar{H}_L^{\ell,\,m}(z) = \sum_{s=\lceil z \rceil}^{s_1} h_L^{\ell,\,m}(s)\,. \tag{2.4}$$

Справедлива формула полной вероятности:

$$\sum_{s=s_0}^{s_1} h_L^{\ell,\,m}(s) = H_L^{\ell,\,m}(s_1) = \bar{H}_L^{\ell,\,m}(s_0) = 1.$$

При фиксированных L, ℓ и m функция $h(s) = h_L^{\ell,\,m}(s)$ является одномерным дискретным распределением. Для примера на рис. 2.1 слева показана функция $h(s)$ при фиксированном $m = 100$. Функция $h'(m) = h_L^{\ell,\,m}(s)$ распределением, вообще говоря, не является, так как не удовлетворяет условию нормировки: $\sum_m h'(m) \neq 1$. На рис. 2.1 снизу показана функция $h'(m)$ при фиксированном $s = 50$.

3. Параметры ℓ и m можно переставлять местами: $h_L^{\ell,\,m}(s) = h_L^{m,\,\ell}(s)$.

4. Параметры m и s можно заменять разностями: $h_L^{\ell,\,m}(s) = h_L^{\ell,\,L-m}(\ell - s)$.

5. Справедливы тождества:

$$h_L^{\ell,\,m}(s) = h_L^{\ell,\,L-m}(\ell - s) = h_L^{m,\,\ell}(s) = h_L^{m,\,k}(m - s) = h_L^{k,\,m}(m - s).$$

6. Отсюда вытекают тождества для функций H и $\bar{H}$:

$$H_L^{\ell,\,m}(s) = \sum_{j=s_0}^{s} h_L^{\ell,\,m}(j) = \sum_{j=s_0}^{s} h_L^{k,\,m}(m - j) = \bar{H}_L^{k,\,m}(m - s).$$

7. Распределение $h(s)$ является унимодальным (имеет форму пика). Максимальное значение достигается при $s^* = \frac{(m+1)(\ell+1)}{L+2}$, с точностью до округления.

8. Таблица гипергеометрического распределения содержит ℓk ненулевых значений. Её можно эффективно вычислить с помощью рекуррентных соотношений:

$$\begin{aligned}
h_L^{\ell,\,0}(0) &= 1;\\
h_L^{\ell,\,m+1}(s) &= h_L^{\ell,\,m}(s)\,\frac{m+1}{m+1-s}\cdot\frac{k-m+s}{L-m};\\
h_L^{\ell,\,m}(s+1) &= h_L^{\ell,\,m}(s)\,\frac{m-s}{s+1}\cdot\frac{\ell-s}{k-m+s+1};\\
h_L^{\ell,\,m}(s-1) &= h_L^{\ell,\,m}(s)\,\frac{s}{m-s+1}\cdot\frac{k-m+s}{\ell-s+1}.
\end{aligned} \tag{2.5}$$

Чтобы избежать вычислительных погрешностей, значения $h_L^{\ell,\,m}(s)$ рекомендуется вычислять последовательно для всех $m = 0, \ldots, L$, при этом для каждого m первым вычислять значение, близкое к максимальному (например, при $s = s_{\max}$), затем меньшие значения вычислять через бóльшие.

9. Математическое ожидание величины s есть

$$\lambda = \sum_{s=s_0}^{s_1} s h_L^{\ell,\,m}(s) = \frac{\ell m}{L}.$$

10. Дисперсия величины s есть

$$\sigma^2 = \sum_{s=s_0}^{s_1} (s-\lambda)^2 h_L^{\ell,\,m}(s) = \lambda\frac{k(L-m)}{L(L-1)}.$$

11. При бóльших значениях параметров L, ℓ, m предельными распределениями для $h(s) = h_L^{\ell,\,m}(s)$ могут быть только распределения одного из четырёх типов:

— при $\lambda \to \infty$ *нормальное распределение* $h(s) \to \frac{1}{\sqrt{2\pi}}\exp\bigl(-\frac{(s-\lambda)^2}{2\sigma^2}\bigr)$;
— при $\frac{m}{L} \to p$ *биномиальное распределение* $h(s) \to C_\ell^s\, p^s(1-p)^{\ell-s}$;
при $\frac{\ell}{L} \to p$ *биномиальное распределение* $h(s) \to C_m^s\, p^s(1-p)^{m-s}$;
— при $\frac{m}{L} \to 0$ или $\frac{\ell}{L} \to 0$ *распределение Пуассона* $h(s) \to e^{-\lambda}\lambda^s/s!$;
— при $\lambda \to 0$ вырожденное распределение $s = 0$.

12. Гипергеометрическое распределение довольно точно приближается с помощью аппроксимации Моленара:

$$h(s) \approx C_\ell^s\,\tilde{p}^s(1-\tilde{p})^{\ell-s}, \quad \tilde{p} = \frac{2m-s}{2L-\ell+1}.$$

§2.3 Закон больших чисел в слабой аксиоматике

Продолжим рассмотрение задач (2.1), (2.2) о предсказании частоты события.

Теорема 2.2. *Пусть $n(\mathbb{X}) = m$. Для любого $\varepsilon \in [0,1)$ справедливы точные оценки:*

$$\mathsf{P}\big[\nu(X) \leqslant \varepsilon\big] = H_L^{\ell,\,m}(\varepsilon\ell)\,; \tag{2.6}$$

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = H_L^{\ell,\,m}(m - \varepsilon k)\,; \tag{2.7}$$

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] = H_L^{\ell,\,m}(s_m(\varepsilon))\,, \qquad s_m(\varepsilon) = \tfrac{\ell}{L}(m - \varepsilon k); \tag{2.8}$$

$$\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] = H_L^{\ell,\,m}(s_m(\varepsilon)) + \bar{H}_L^{\ell,\,m}(\bar{s}_m(\varepsilon))\,, \quad \bar{s}_m(\varepsilon) = \tfrac{\ell}{L}(m + \varepsilon k). \tag{2.9}$$

Доказательство. Первые две оценки являются непосредственным следствием (2.3):

$$\mathsf{P}\big[\nu(X) \leqslant \varepsilon\big] = \sum_{s=0}^{\ell} \big[\tfrac{s}{\ell} \leqslant \varepsilon\big]\, \mathsf{P}\big[n(X) = s\big] = \sum_{s=s_0}^{\lfloor \varepsilon\ell \rfloor} h_L^{\ell,\,m}(s) = H_L^{\ell,\,m}(\varepsilon\ell)\,;$$

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = \sum_{s=0}^{\ell} \big[\tfrac{m-s}{k} \geqslant \varepsilon\big]\, \mathsf{P}\big[n(\bar{X}) = m - s\big] = \sum_{s=s_0}^{\lfloor m - \varepsilon k \rfloor} h_L^{\ell,\,m}(s) = H_L^{\ell,\,m}(m - \varepsilon k)\,.$$

Третья оценка получается аналогично первой, с той лишь разницей, что условие $\nu(X)\frac{s}{\ell} \leqslant \varepsilon$ заменяется условием $\nu(\bar{X}) - \nu(X) = \frac{m-s}{k} - \frac{s}{\ell} \geqslant \varepsilon$. Отсюда элементарными преобразованиями получаем верхний предел суммирования $s \leqslant \frac{\ell}{L}(m - \varepsilon k)$.

Наконец, двусторонняя оценка (2.9) получается, если представить множество разбиений в виде объединения двух непересекающихся подмножеств:

$$\begin{aligned}\mathsf{P}\big[|\nu(\bar{X}) - \nu(X)| \geqslant \varepsilon\big] &= \mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] + \mathsf{P}\big[\nu(X) - \nu(\bar{X}) \geqslant \varepsilon\big] = \\ &= H_L^{\ell,\,m}(s_m(\varepsilon)) + \bar{H}_L^{\ell,\,m}(\bar{s}_m(\varepsilon)).\end{aligned}$$

■

О скорости сходимости в законе больших чисел. При пропорциональном увеличении L, ℓ и m относительная ширина гипергеометрического пика уменьшается, рис. 2.2. В пределе при $L, \ell, m \to \infty$ возможно сколь угодно точное предсказание скрытой частоты $\nu(\bar{X})$ по наблюдаемой частоте $\nu(X)$. Равенство (2.9) оценивает скорость сходимости частот события в двух выборках.

Классический закон больших чисел утверждает сходимость частоты события к её вероятности. Однако в слабой аксиоматике понятие «вероятности события S» не определено. Поэтому (2.9) можно интерпретировать как *аналог закона больших чисел* в слабой аксиоматике. Основанием для такой интерпретации также служит тот факт, что два функционала — (а) вероятность большого отклонения частот события в двух выборках и (б) вероятность большого отклонения частоты события от его вероятности — оцениваются сверху друг через друга, как показано в [6]. По сути, эти две оценки отличаются не принципиально.

Классические неравенства Чебышёва, Чернова, Бернштейна, Хёффдинга [45] оценивают скорость сходимости в законе больших чисел. Все они являются асимптотическими и дают завышенные оценки вероятности большого отклонения. Выражение (2.9) является точной (не завышенной, не асимптотической) оценкой, и потому его можно считать наиболее точным выражением закона больших чисел.

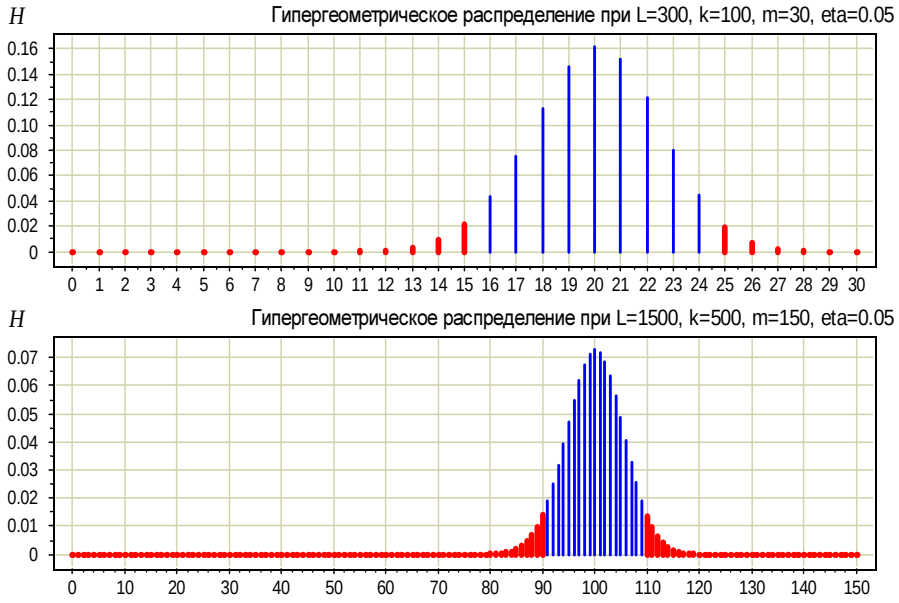


Рис. 2.2. Гипергеометрическая функция $h_L^{\ell,m}(s)$. Верхний график получен при $L = 300$, $\ell = 200$, $m = 30$. Выделены крайние левые, $[s_0, s_m(\varepsilon)] = [0, 15]$, и крайние правые $[s_m(\varepsilon), s_1] = [25, 30]$, члены распределения, соответствующие значению надёжности $\eta = 0.05$. Нижний график получен при значениях L, ℓ, m, пропорционально увеличенных в 5 раз.

Геометрическая интерпретация соотношений (2.3), (2.8) и (2.9). Рассмотрим прямоугольную сетку $\{0, \ldots, L\} \times \{0, \ldots, \ell\}$ на рис. 2.3. Положим $b_i = [x_i \in X]$. Тогда $b_i = 1$ означает, что объект x_i попадает в наблюдаемую подвыборку. Договоримся отображать разбиение $X, \bar{X}$ в виде траектории, проходящей по узлам сетки из точки $(0, 0)$ в точку (L, ℓ) согласно правилу: если $b_i = 1$, то смещаемся на единицу вправо-вверх; если $b_i = 0$, то смещаемся на единицу вправо. Все такие траектории не выходят за пределы параллелограмма, выделенного на рис. 2.3. Множество всех таких траекторий изоморфно множеству разбиений выборки $\mathbb{X} = X \sqcup \bar{X}$, и оба они изоморфны множеству L-мерных бинарных векторов $(b_1, \ldots, b_L)$, содержащих ровно ℓ единиц. Поэтому для подсчёта числа разбиений, удовлетворяющих некоторому свойству, достаточно найти число соответствующих траекторий.

Чтобы вывести (2.3), пронумеруем объекты выборки так, чтобы первые m объектов принадлежали множеству S. Тогда задача сведётся к подсчёту доли траекторий, проходящих через точку (m, s). Назовём такие траектории *допустимыми*. Число всех возможных траекторий на отрезке от $(0, 0)$ до (m, s) равно C_m^s, и для каждой траектории существует $C_{L-m}^{\ell-s}$ вариантов её продолжения от (m, s) до (L, ℓ). Следовательно, число допустимых траекторий равно $C_m^s C_{L-m}^{\ell-s}$. Разделив на общее число возможных траекторий C_L^ℓ, получаем требуемое $h_L^{\ell,m}(s)$.

Чтобы вывести (2.8), необходимо подсчитать число траекторий, проходящих через любую точку (m, s), лежащую ниже диагонали на $\varepsilon\frac{\ell k}{L}$ или более. Для этого суммируется число траекторий $C_m^s C_{L-m}^{\ell-s}$ по всем $s = s_0, \ldots, s_m(\varepsilon)$.

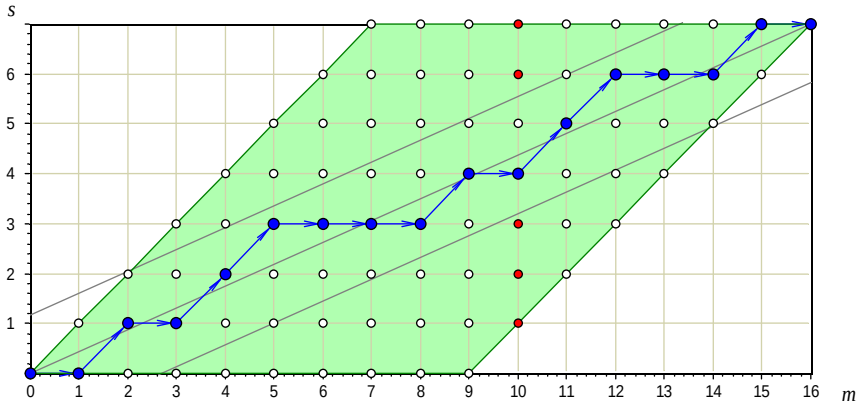


Рис. 2.3. Траектория последовательности $\{b_i\}_{i=0}^{L} = 0101100010110010$, при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$. Проведены линии $s = \frac{\ell}{L}m$ и $s = \frac{\ell}{L}(m \pm \varepsilon k)$. При $m = 10$ выделены точки выше верхней линии, $s \geqslant \bar{s}_m(\varepsilon)$, и ниже нижней линии, $s \leqslant s_m(\varepsilon)$.

Для вывода двусторонней оценки (2.9) подсчитывается число траекторий, отстоящих от диагонали на $\varepsilon\frac{\ell k}{L}$ или более. В этом случае суммирование идёт по всем $s = s_0, \ldots, s_m(\varepsilon)$, затем по всем $s = \bar{s}_m(\varepsilon), \ldots, s_1$.

Выборочный контроль качества — это пример прикладной задачи, в которой оценки Теоремы 2.2 применимы непосредственно [3].

Пусть $\mathbb{X}$ — множество изделий, $S \subset \mathbb{X}$ — подмножество дефектных изделий. Изготовлена партия изделий $\mathbb{X}$, из них m оказались дефектными. Число m неизвестно. Проверить всю партию поштучно не представляется возможным. Поэтому делается *выборочный контроль качества*: случайно, независимо, без возвращений выбирается подмножество $X \subset \mathbb{X}$, что равносильно случайному равномерному выбору разбиения $X \sqcup \bar{X} = \mathbb{X}$. Зная долю дефектов в наблюдаемой выборке $\nu(X)$, требуется предсказать долю дефектов в скрытой выборке $\nu(\bar{X})$. Если при заданной точности ε и надёжности η имеет место оценка $\mathsf{P}[\nu(\bar{X}) \geqslant \varepsilon] < \eta$, то партия $\mathbb{X}$ принимается, иначе она целиком бракуется. Параметры ε и η подбираются из экономических соображений — с учётом стоимости контроля одного изделия и величины потерь от использования дефектного изделия.

§2.4 Переход от ненаблюдаемой оценки к наблюдаемой

Оценки (2.6)–(2.9) зависят от числа элементов m события S в генеральной выборке $\mathbb{X}$, которое невозможно узнать, пока неизвестна скрытая подвыборка. В таких случаях говорят, что оценка является *ненаблюдаемой* (unobservable bound). Ситуация на первый взгляд парадоксальна. Чтобы оценить вероятность большого отклонения частот в наблюдаемой и скрытой выборке, необходимо знать число m. Однако если бы мы его знали, то по наблюдаемой частоте $\nu(X)$ тут же вычисли бы точное значение скрытой частоты $\nu(\bar{X})$, поскольку $k\nu(\bar{X}) + \ell\nu(X) = m$.

Верхняя оценка. Простейшее решение проблемы неизвестного m заключается в том, чтобы взять максимум по m и получить вместо точной оценки завышенную

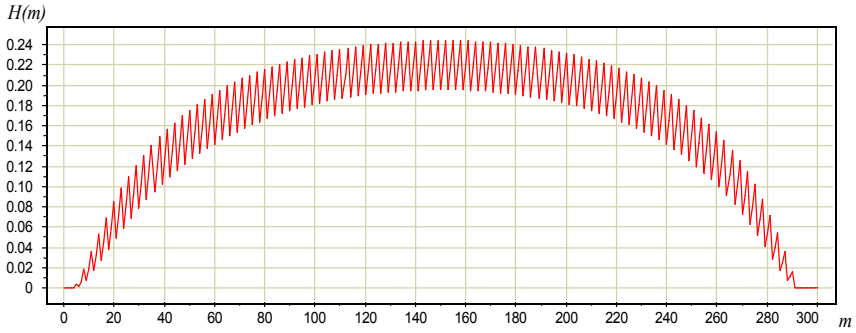


Рис. 2.4. Зависимость $H(m) = H_L^{\ell,\,m}\,(s_m(\varepsilon))$ от m при $L = 300,\ \ell = 200,\ \varepsilon = 0.05$.

верхнюю оценку:

$$\mathsf{P}\big[\nu(\bar{X}) - \nu(X) \geqslant \varepsilon\big] \leqslant \max_{m=0,\ldots,L} H_L^{\ell,\,m}\,(s_m(\varepsilon)) \equiv \Gamma_L^{\ell}(\varepsilon). \tag{2.10}$$

Здесь максимум достаточно взять по всем m от $\lceil \varepsilon k \rceil$ до $\lfloor L - \varepsilon\ell \rfloor$, так как при остальных значениях m левая часть неравенства равна нулю.

К сожалению, (2.10) — довольно грубая оценка при малых m, см. рис. 2.4. По этой причине данный подход не приемлем для задач выборочного контроля качества, обучения по прецедентам, и других случаев, когда именно малые значения m представляют большой практический интерес.

Известна верхняя оценка «хвоста» гипергеометрического распределения [36], с помощью которой можно оценить сверху правую часть (2.10): для любого $\varepsilon > 0$

$$\Gamma_L^{\ell}(\varepsilon) \leqslant \exp\left(-2\varepsilon^2 \frac{\ell k^2}{L^2}\right).$$

Эта оценка ещё более грубая. На рис. 2.4 ей соответствует горизонтальная линия с ординатой 0.89, но она не показана, поскольку проходит много выше. Асимптотически эта оценка сходится к нулю при одновременном стремлении ℓ и k к бесконечности, что ещё раз подтверждает связь точных оценок (2.8) и (2.9) с *законом больших чисел*.

Точная интервальная оценка. Следующая теорема показывает, как получать точные верхние и нижние оценки для $n(\mathbb{X})$ и $n(\bar{X})$ по наблюдаемому значению $s = n(X)$.

Теорема 2.3. *Если $s = n(X)$ — число элементов события S в наблюдаемой выборке, то для числа элементов события S в полной выборке с вероятностью $(1 - \eta)$ справедлива верхняя оценка:*

$$n(\mathbb{X}) \leqslant \max\big\{m = m_0, \ldots, L \;\big|\; H_L^{\ell,\,m}\,(s) \geqslant \eta\big\}, \quad \text{где}\ \ m_0 = \big\lceil s\tfrac{L+2}{\ell+1} - 1\big\rceil. \tag{2.11}$$

Доказательство. Рассмотрим одностороннюю точную оценку (2.7), обозначив правую её часть через $H(\varepsilon, m)$, где $m = n(\mathbb{X})$ — неизвестная величина:

$$\mathsf{P}\big[\nu(\bar{X}) \geqslant \varepsilon\big] = H_L^{\ell,\,m}\,(m - \varepsilon k) = H(\varepsilon, m). \tag{2.12}$$

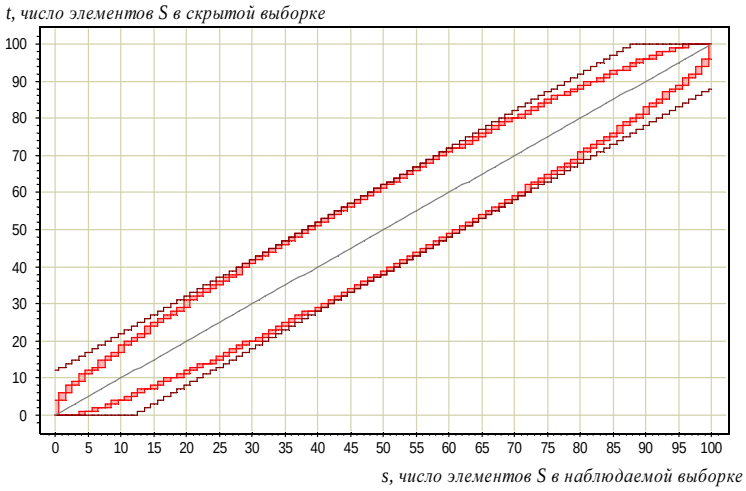


Рис. 2.5. Точные верхние и нижние оценки числа $t = n(\bar{X})$ элементов события S в скрытой выборке в зависимости от их числа $s = n(X)$ в наблюдаемой выборке. Условия эксперимента: $L = 200,\ \ell = k = 100,\ \eta = 0.05$.

Тогда с вероятностью $(1-\eta)$ справедлива оценка сверху $\nu(\bar{X}) < E(\eta, m)$, где $E(\eta, m)$ — обратная функция от $H(\varepsilon, m)$. Обращение производится по первому аргументу при каждом значении второго аргумента m, который выступает в роли параметра. Поскольку функция $E(\eta, m)$ не возрастает по первому аргументу, из оценки $\nu(\bar{X}) < E(\eta, m)$ следует неравенство $H(\nu(\bar{X}), m) \geqslant \eta$. Подставляя $\nu(\bar{X}) = \frac{m-s}{k}$ в функцию $H(\nu(\bar{X}), m)$, определяемую согласно (2.12), получаем, что с вероятностью $(1-\eta)$ справедливо неравенство $H_L^{\ell,\,m}(s) \geqslant \eta$. Чтобы разрешить данное неравенство относительно m при фиксированном s, найдём максимальное значение m, при котором оно выполнено. При максимальном значении m значение s должно находится левее точки максимума гипергеометрического распределения $s^* = \frac{(m+1)(\ell+1)}{L+2}$, Следовательно, $s(L+2) < (m+1)(\ell+1)$. Поэтому для нахождения максимального значения m достаточно перебрать значения m, не меньшие $s\frac{L+2}{\ell+1} - 1$. ■

Следствие 2.3.1. *Аналогично оценивается скрытое число $n(\bar{X})$ по наблюдаемому числу $s = n(X)$: с вероятностью $(1-\eta)$ выполнено неравенство*

$$n(\bar{X}) \leqslant \max\bigl\{t = t_0, \ldots, k \;\bigm|\; H_L^{\ell,\,s+t}(s) \geqslant \eta\bigr\}, \quad t_0 = \bigl\lceil s\tfrac{k+1}{\ell+1} - 1\bigr\rceil. \tag{2.13}$$

Следствие 2.3.2. *Аналогично строятся и нижние оценки: с вероятностью $(1-\eta)$*

$$n(\mathbb{X}) \geqslant \min\bigl\{m = 0, \ldots, m_0 \;\bigm|\; \bar{H}_L^{\ell,\,m}(s) \geqslant \eta\bigr\};$$
$$n(\bar{X}) \geqslant \min\bigl\{t = 0, \ldots, t_0 \;\bigm|\; \bar{H}_L^{\ell,\,s+t}(s) \geqslant \eta\bigr\},$$

Вычисление полученных верхних и нижних оценок с использованием рекуррентных соотношений (2.5) требует порядка $O\bigl(n(X)n(\bar{X})\bigr)$ операций.

На рис. 2.5 показаны верхние и нижние оценки числа элементов события S в скрытой выборке $t = n(\bar{X})$ в зависимости от их числа в наблюдаемой выборке $s = n(X)$. Толстые ступенчатые линии — граничные области, в которых выполняется равенство $H_L^{\ell,\,s+t}(s) = \eta$. Точная верхняя оценка совпадает с верхней границей

верхней области, точная нижняя — с нижней границей нижней области. Вместе они определяют $1 - 2\eta = 90\%$-й доверительный интервал для числа t при каждом значении s. Тонкие ступенчатые линии — это оценки по наихудшему m, вычисленные согласно (2.10). Их точность падает по мере приближения m к 0 или к L.

О вероятности нуль-события. Пользуясь Теоремой 2.3, нетрудно посчитать верхнюю доверительную оценку $n(\mathbb{X})$ для *нуль-события* [15] — такого события, которое вообще не наблюдалось, $n(X) = 0$. Данная задача имеет точное решение (2.13), в которое надо подставить $s = 0$. В частности, по графику на рис. 2.5 легко определить, что при $s = 0$ и длине наблюдаемой выборки $\ell = 100$ число событий в скрытой выборке длины $k = 100$ не превзойдёт 4 с вероятностью $1 - \eta = 95\%$. Нижняя доверительная оценка $n(\mathbb{X})$ для нуль-события, разумеется, равна нулю.

§2.5 Одноэлементное семейство алгоритмов

Вернёмся к задаче оценивания вероятности переобучения и рассмотрим одноэлементное семейство алгоритмов $A = \{a\}$. В этом случае никакого обучения быть не может: $\mu X = a$ для любой выборки X. Функционал $Q_\varepsilon(\mu, \mathbb{X}) \equiv Q_\varepsilon(a, \mathbb{X})$ будем называть *вероятностью переобучения отдельного алгоритма*.

Теорема 2.4. *Пусть алгоритм a допускает m ошибок на генеральной выборке: $n(a, \mathbb{X}) = m$. Тогда для любого $\varepsilon \in [0, 1]$ справедливы точные оценки:*

$$Q_\varepsilon = \mathsf{P}\big[\nu(a, \bar{X}) - \nu(a, X) \geqslant \varepsilon\big] = H_L^{\ell, m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right); \tag{2.14}$$

$$R_\varepsilon = \mathsf{P}\big[\nu(a, \bar{X}) \geqslant \varepsilon\big] = H_L^{\ell, m}\left(m - \varepsilon k\right); \tag{2.15}$$

$$C = \mathsf{E}\nu(a, \bar{X}) = \tfrac{m}{L}. \tag{2.16}$$

Доказательство. Равенства (2.14) и (2.15) вытекают непосредственно из Теоремы 2.2, если ввести событие $S = \big\{x_i \in \mathbb{X} \colon I(a, x_i) = 1\big\}$. Для доказательства (2.16) запишем определения E и ν, затем переставим знаки суммирования:

$$C = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \frac{1}{k} \sum_{x_i \in \bar{X}} I(a, x_i) = \frac{1}{kC_L^\ell} \underbrace{\sum_{i=1}^{L} I(a, x_i)}_{m} \underbrace{\sum_{X \in [\mathbb{X}]^\ell} [x_i \in \bar{X}]}_{C_{L-1}^\ell} = \frac{mC_{L-1}^\ell}{kC_L^\ell} = \frac{m}{L}.$$

■

Резюме

Задача оценивания частоты события (например, ошибки фиксированного алгоритма a) на скрытой выборке по его частоте на наблюдаемой выборке имеет точное решение, которое выражается через гипергеометрическое распределение.

В слабой вероятностной аксиоматике эта оценка описывает скорость сходимости частот в двух выборках и является аналогом закона больших чисел.

Неудобство этой оценки заключается в том, что она зависит от неизвестной частоты события на генеральной выборке m. Проблема решается либо взятием максимума по m, и тогда получается завышенная оценка, либо вычислением точной интервальной оценки, но тогда она не выражается в виде явной формулы.

В следующей лекции мы перейдём к оценкам вероятности переобучения для произвольных семейств алгоритмов и произвольных методов обучения μ. В этом случае получение точных оценок становится довольно сложной задачей. Мы начнём с верхних оценок Вапника-Червоненкиса, которые исторически были первыми. Они относительно несложно выводятся, но сильно завышены.

Упражнения

Задача 2.1 (1). Построить графики зависимости величины Q_ε, определяемой по формуле (2.14), от ℓ при фиксированном k, от k при фиксированном ℓ и от $\ell = k$.

Задача 2.2 (2). Вывести точную оценку вероятности переобучения Q_ε для семейства из двух алгоритмов $A = \{a_1, a_2\}$, если заданы четыре параметра

$$m_{rs} = \#\big\{x \in \mathbb{X}\colon I(a_1, x) = r,\ I(a_2, x) = s\big\}, \qquad r, s \in \{0, 1\}.$$

Построить графики зависимости Q_ε от хэммингова расстояния между алгоритмами $\rho_{12} = m_{01} + m_{10}$, в двух случаях: 1) при $m_{01} = m_{10}$; 2) при $m_{01} = 0$.

Задача 2.3 (3). Вывести точную оценку вероятности переобучения Q_ε для семейства из трёх алгоритмов $A = \{a_1, a_2, a_3\}$, если заданы восемь параметров

$$m_{rst} = \#\big\{x \in \mathbb{X}\colon I(a_1, x) = r,\ I(a_2, x) = s,\ I(a_3, x) = t\big\}, \qquad r, s, t \in \{0, 1\}.$$

Построить графики зависимости Q_ε от хэммингова расстояния между алгоритмами $\rho_{12} = m_{010} + m_{100} + m_{011} + m_{101}$, в двух случаях: 1) при $m_{110} = m_{101} = m_{011} = m_{100} = m_{010} = m_{001}$; 2) при $m_{110} = m_{101} = m_{100} = m_{010} = 0$, $m_{011} = m_{001}$.

3 Теория Вапника-Червоненкиса

Статистическая теория восстановления зависимостей по эмпирическим данным (VC-теория) была предложена В. Н. Вапником и А. Я. Червоненкисом в конце 60-х — начале 70-х годов [7, 8, 9, 6]. В середине 80-х она получила широкую мировую известность [55, 56, 57] и вместе с работами Валианта [54] на многие годы определила генеральное направление развития теории статистического обучения.

Рассмотрим основные предположения и результаты VC-теории в рамках слабой вероятностной аксиоматики, используя обозначения предыдущей главы.

§3.1 Коэффициенты разнообразия и профиль расслоения

Напомним, что $\mathbb{X} = \{x_1, \ldots, x_L\}$ — конечное множество объектов, A — множество алгоритмов, $I(a,x)$ — индикатор ошибки. Каждый алгоритм $a \in A$ порождает на $\mathbb{X}$ вектор ошибок $\vec{a} = \big(I(a,x_i)\big)_{i=1}^{L}$. Введём ещё несколько понятий и обозначений.

$\vec{A} = \big\{\vec{a} \colon a \in A\big\}$ — множество векторов ошибок, порождаемых множеством алгоритмов A на заданной выборке $\mathbb{X}$. Мощность $|\vec{A}|$ конечна, не превышает мощности множества A и не превышает 2^L — числа различных булевых векторов длины L.

$\Delta(A,\mathbb{X}) = |\vec{A}|$ — *коэффициент разнообразия* (shattering coefficient)[4] множества алгоритмов A на выборке $\mathbb{X}$. В задачах классификации на два класса коэффициент разнообразия равен числу различных *дихотомий* (способов разделить выборку $\mathbb{X}$ на два класса), реализуемых всевозможными алгоритмами из A.

$A_L^\ell \equiv A_L^\ell(\mu,\mathbb{X}) = \big\{\mu X \colon X \subset \mathbb{X},\ |X| = \ell\big\}$ — множество алгоритмов, индуцируемых методом обучения μ на всевозможных обучающих подвыборках X. Мощность $|A_L^\ell|$ конечна и не превышает C_L^ℓ — числа различных разбиений $X \sqcup \bar{X} = \mathbb{X}$.

$\Delta_L^\ell \equiv \Delta_L^\ell(\mu,\mathbb{X}) = \Delta\big(A_L^\ell(\mu,\mathbb{X}),\mathbb{X}\big)$ — *локальный коэффициент разнообразия* (local shatter coefficient) метода μ на выборке $\mathbb{X}$. Локальный коэффициент разнообразия не превосходит C_L^ℓ. Он может оказаться и строго меньше C_L^ℓ, поскольку метод μ может строить по различным выборкам совпадающие алгоритмы; кроме того, различные алгоритмы могут порождать одинаковые векторы ошибок.

$\Delta^A(L) = \max\limits_{\mathbb{X}} \Delta(A,\mathbb{X})$ — *глобальный коэффициент разнообразия* (global shatter coefficient), называемый также *функцией роста* (growth function) множества алгоритмов A [9, 57]. Максимум берётся по всевозможным выборкам $\mathbb{X} \subset \mathscr{X}$ длины L из некоторого (как правило, бесконечного) множества допустимых объектов $\mathscr{X}$. Функция роста является мерой сложности множества алгоритмов A. В отличие от локального

[4] В исходных работах В. Н. Вапника и А. Я. Червоненкиса [8, 9, 6] коэффициент разнообразия назывался *индексом системы событий*. Алгоритм a индуцирует событие $S_a = \big\{x \in \mathbb{X} \;\big|\; I(a,x) = 1\big\}$. Семейство A индуцирует систему событий $S = \{S_a \,|\, a \in A\}$. Индекс системы событий S есть число различных подмножеств вида $S_a \cap \mathbb{X}$, где a пробегает всё множество A, что равносильно определению через $|\vec{A}|$. В англоязычных работах прижился термин shattering — число разбиений всеми возможными способами, буквально «вдребезги». Другой вариант перевода — «дробление» [28].

коэффициента разнообразия, она не зависит ни от задачи (выборки $\mathbb{X}$ и восстанавливаемой зависимости $y(x)$), ни от метода обучения μ. Поэтому $\Delta^A(L)$ может оказаться существенно больше $\Delta_L^\ell(\mu,\mathbb{X})$. Справедлива верхняя оценка $\Delta^A(L) \leqslant 2^L$.

$A_m = \{a \in A\colon n(a,\mathbb{X}) = m\}$ — множество алгоритмов из A с m ошибками на генеральной выборке $\mathbb{X}$. Будем называть подмножества A_m *слоями* и говорить, что A *расслаивается по уровням ошибок*. Очевидно, $A = A_0 \sqcup \cdots \sqcup A_L$.

$\Delta_m \equiv \Delta_m(\mu,\mathbb{X}) = \Delta\big((A_L^\ell)_m,\mathbb{X}\big)$ — локальный коэффициент разнообразия m-го слоя множества алгоритмов $A_L^\ell(\mu,\mathbb{X})$. Совокупность величин $(\Delta_m)_{m=0}^L$ будем называть *профилем расслоения*[5]. Очевидно, $\Delta_L^\ell = \Delta_0 + \cdots + \Delta_L$.

§3.2 Оценка Вапника-Червоненкиса

Принцип равномерной сходимости вводится в VC-теории и многих последующих работах (см. обзоры [58, 34, 10]), чтобы получать верхние оценки вероятности переобучения, не зависящие от метода μ. Принцип заключается в том, чтобы заменить функционал Q_ε его верхней оценкой $\widetilde{Q}_\varepsilon$ — вероятностью большого равномерного отклонения частот в двух подвыборках:

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon = \mathsf{P}\Big[\max_{a\in A_L^\ell}\delta(a,X) \geqslant \varepsilon\Big] = \mathsf{P}\max_{a\in A_L^\ell}\big[\delta(a,X) \geqslant \varepsilon\big], \tag{3.1}$$

где $\delta(a,X) = \nu(a,\bar{X}) - \nu(a,X)$. Термин *сходимость* означает, что если $\widetilde{Q}_\varepsilon \to 0$ при $\ell,k\to\infty$ для любого $\varepsilon \in (0,1)$, то переобученность также сходится к нулю: $\delta(a,X)\to 0$. Термин *равномерность* означает, что величина $\delta(a,X)$ сходится к нулю одновременно для всех алгоритмов $a \in A_L^\ell$, в том числе для алгоритма $a = \mu X$, какими бы ни были метод обучения μ и выборка X.

О завышенности оценки равномерной сходимости. Разумеется, замена величины $\delta(a,X)$ её максимумом по всему семейству $a \in A$ является грубой оценкой. Тем не менее, она может оказаться точной даже в тех случаях, когда мощность $|A|$ очень велика. Следующая теорема показывает, что неравенство (3.1) обращается в равенство, когда множество алгоритмов A_L^ℓ *не расслаивается* по уровням ошибок.

Теорема 3.1. *Если метод μ минимизирует эмпирический риск, и все алгоритмы $a \in A_L^\ell$ имеют одинаковый уровень ошибок $m = n(a,\mathbb{X})$, то верхняя оценка* (3.1) *обращается в точное равенство: $Q_\varepsilon = \widetilde{Q}_\varepsilon$.*

Доказательство. Минимизация эмпирического риска $\nu(a,X)$ при фиксированном m эквивалентна максимизации переобученности, поскольку

$$\delta(a,X) = \frac{m - \ell\nu(a,X)}{k} - \nu(a,X) = \frac{m}{k} - \frac{L}{k}\nu(a,X).$$

Теорема доказана. ∎

[5] В статье [59] она называлась *профилем разнообразия* множества алгоритмов A на выборке $\mathbb{X}$.

Если же множество A_L^ℓ расслаивается по уровням ошибок, то (3.1) может оказаться как точной, так и сильно завышенной верхней оценкой, что будет показано в дальнейшем на примерах. В общем случае требование равномерной сходимости является избыточно сильным и даёт лишь достаточное условие обучаемости.

Основная теорема VC-теории. Здесь приводится доказательство, существенно более краткое, чем в исходных работах [8, 9, 6].

Теорема 3.2. *Для любых μ, $\mathbb{X}$ и $\varepsilon \in [0,1]$ справедлива оценка*

$$Q_\varepsilon \leqslant \Delta_L^\ell(\mu,\mathbb{X}) \max_{m=1,\ldots,L} H_L^{\ell,\,m}\left(s_m(\varepsilon)\right), \quad s_m(\varepsilon) = \tfrac{\ell}{L}(m-\varepsilon k). \tag{3.2}$$

Доказательство. Воспользуемся принципом равномерной сходимости и заметим, что максимум в (3.1) достаточно взять только по алгоритмам, неразличимым на выборке $\mathbb{X}$, т. е. по множеству векторов ошибок $\vec{A}_L^\ell$:

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon = \mathsf{P} \max_{\vec{a}\in\vec{A}_L^\ell} \big[\delta(a,X) \geqslant \varepsilon\big]. \tag{3.3}$$

Оценим максимум бинарных величин их суммой[6], переставим местами знаки суммирования: $\mathsf{P}\sum = \sum \mathsf{P}$ и запишем вероятность $\mathsf{P}\big[\delta(a,X) \geqslant \varepsilon\big]$ для отдельного алгоритма a согласно Теореме 2.4:

$$\widetilde{Q}_\varepsilon \leqslant \widetilde{\widetilde{Q}}_\varepsilon = \sum_{\vec{a}\in\vec{A}_L^\ell} \mathsf{P}\big[\delta(a,X) \geqslant \varepsilon\big] = \sum_{\vec{a}\in\vec{A}_L^\ell} H_L^{\ell,\,m}\left(s_m(\varepsilon)\right), \quad m = n(a,\mathbb{X}). \tag{3.4}$$

Представим сумму по $\vec{a} \in \vec{A}_L^\ell$ двойной суммой: сначала по слоям m, затем по алгоритмам m-го слоя $\vec{a} \in (\vec{A}_L^\ell)_m$. Тогда

$$\widetilde{\widetilde{Q}}_\varepsilon = \sum_{m=0}^{L} \Delta_m H_L^{\ell,\,m}\left(s_m(\varepsilon)\right). \tag{3.5}$$

Оценим сомножитель $H_L^{\ell,\,m}\left(s_m(\varepsilon)\right)$ сверху максимумом по m, вынесем его за знак суммы по m и воспользуемся равенством $\Delta_L^\ell = \Delta_0 + \cdots + \Delta_L$:

$$Q_\varepsilon \leqslant \widetilde{Q}_\varepsilon \leqslant \widetilde{\widetilde{Q}}_\varepsilon \leqslant \Delta_L^\ell \max_m H_L^{\ell,\,m}\left(s_m(\varepsilon)\right). \tag{3.6}$$

Теорема доказана. ■

Полностью аналогично доказывается верхняя оценка и для функционала R_ε.

Следствие 3.2.1. *Для любых μ, $\mathbb{X}$ и $\varepsilon \in [0,1]$ справедлива оценка*

$$R_\varepsilon \leqslant \sum_{\vec{a}\in\vec{A}_L^\ell} H_L^{\ell,\,m}\left(m-\varepsilon k\right) \;\leqslant\; \Delta_L^\ell(\mu,\mathbb{X}) \max_{m=1,\ldots,L} H_L^{\ell,\,m}\left(m-\varepsilon k\right).$$

[6] Это можно трактовать и как оценку вероятности объединения событий $\big[\delta(a,X) \geqslant \varepsilon\big]$ сверху суммой их вероятностей. Её называют также *неравенством Буля* или union bound.

Таким образом, имеется серия *VC-оценок*.

Наиболее точная оценка (3.4) — это сумма вероятностей переобучения по всем алгоритмам семейства с попарно различными векторами ошибок.

Оценка (3.2) чуть хуже — это вероятность переобучения наихудшего алгоритма (максимум $H_L^{\ell,\,m}(s_m(\varepsilon))$ достигается при m порядка $L/2$), помноженная на число алгоритмов с попарно различными векторами ошибок.

В исходных работах [8, 9] вместо (3.1) использовалась более грубая оценка, не зависящая от выборки и метода обучения — максимум брался по всем алгоритмам семейства A, включая и те, которые никогда не выбираются методом обучения. Легко понять, что замена A_L^ℓ на A в функционале $\widetilde{Q}_\varepsilon$ приводит к замене в (3.2) локального коэффициента разнообразия на функцию роста:

$$\Delta_L^\ell(\mu, \mathbb{X}) \leqslant \Delta^A(L). \tag{3.7}$$

Наконец, функцию гипергеометрического распределения можно заменить экспоненциальной верхней оценкой, имеющей особо простой вид при $\ell = k$ [6]:

$$\max_m H_L^{\ell,\,m}(s_m(\varepsilon)) \leqslant \tfrac{3}{2} e^{-\varepsilon^2 \ell}, \quad \ell = k. \tag{3.8}$$

В итоге получается наиболее известная из VC-оценок [57]:

Следствие 3.2.2. *Для любых μ, $\mathbb{X}$, $\varepsilon \in [0,1]$ при $\ell = k$ справедлива оценка:*

$$Q_\varepsilon \leqslant \Delta^A(2\ell) \cdot \tfrac{3}{2} e^{-\varepsilon^2 \ell}. \tag{3.9}$$

Корректные методы обучения. В VC-теории [6] отдельно рассматривается так называемая *детерминистская постановка задачи*[7], когда метод μ *корректен*, то есть $n(\mu X, X) = 0$ на любой обучающей выборке X.

Теорема 3.3. *Если метод μ корректен, то для любых $\mathbb{X}$, $\varepsilon \in [0,1]$*

$$Q_\varepsilon \leqslant \sum_{m=\lceil \varepsilon k \rceil}^{L} \Delta_m \frac{C_{L-m}^\ell}{C_L^\ell} \leqslant \Delta_L^\ell \frac{C_{L-\lceil \varepsilon k \rceil}^\ell}{C_L^\ell} \leqslant \Delta^A(L) \left(\frac{k}{L}\right)^{\varepsilon k}. \tag{3.10}$$

Доказательство. Начало доказательства в точности повторяет доказательство Теоремы 3.2, но к выражению для вероятности переобучения алгоритма a добавляется условие корректности:

$$\mathsf{P}\big[\delta(a,X) \geqslant \varepsilon\big]\big[n(a,X) = 0\big] = \big[m \geqslant \varepsilon k\big]\,\mathsf{P}\big[n(a,X) = 0\big] = [m \geqslant \varepsilon k]\, h_L^{\ell,\,m}(0)\,.$$

[7] В зарубежной литературе сложилась другая терминология. Детерминистскую постановку задачи называют *реализуемым обучением* (realizable learning), имея в виду, что с помощью семейства алгоритмов A возможно реализовать истинную зависимость $y(x)$. Общую постановку задачи называют нереализуемым или *агностическим обучением* (agnostic learning), подчёркивая принципиальную невозможность знать, находится ли истинная зависимость в семействе A, или нет.

Подставим это выражение в (3.4) и, с учётом $h_L^{\ell,m}(0) = C_{L-m}^{\ell}/C_L^{\ell}$, получим

$$Q_\varepsilon \leqslant \widetilde{\widetilde{Q}}_\varepsilon = \sum_{m=\lceil \varepsilon k \rceil}^{L} \Delta_m \frac{C_{L-m}^{\ell}}{C_L^{\ell}}.$$

Максимум $C_{L-m}^{\ell}/C_L^{\ell}$ достигается при наименьшем $m = \lceil \varepsilon k \rceil$. С учётом оценок $\Delta_L^{\ell} \leqslant \Delta^A(L)$ и $C_{L-m}^{\ell}/C_L^{\ell} \leqslant \left(\frac{k}{L}\right)^m$, получаем цепочку неравенств (3.10). ■

Следствие 3.3.1. *Если метод μ корректен, то для любых $\mathbb{X}$, $\varepsilon \in [0,1]$ при $\ell = k$*

$$Q_\varepsilon \leqslant \Delta^A(2\ell) \cdot 2^{-\varepsilon k}.$$

Таким образом, в случае корректности оценка Q_ε становится существенно более точной и принимает наиболее простой вид. Однако отсюда совершенно не следует, что на практике надо пользоваться корректными методами обучения. Для обеспечения корректности придётся усложнять конструкцию семейства алгоритмов, что может привести к увеличению функции роста $\Delta^A(L)$, настолько значительному, что оно полностью скомпенсирует уменьшение комбинаторного множителя. Поэтому в VC-теории принято считать, что не следует добиваться безошибочной работы алгоритма на обучающем материале. С другой стороны, усложнение конструкции алгоритмов может и не увеличивать *локальный* коэффициент разнообразия $\Delta_L^{\ell}(\mu, \mathbb{X})$. Таким образом, вопрос о влиянии корректности на вероятность переобучения средствами VC-теории, по всей видимости, не решается.

§3.3 Метод структурной минимизации риска

Функция роста не зависит ни от выборки, ни от метода обучения, и является мерой сложности множества алгоритмов A. На следующей лекции мы выясним, что с ней тесно связана ещё одна характеристика сложности, называемая *ёмкостью* или *размерностью Вапника-Червоненкиса* (VC-dimension). Пока мы не будем давать её определение, а упомянем лишь два факта. Во-первых, ёмкость семейства линейных классификаторов равна в точности размерности пространства признаков. Во-вторых, если семейство имеет ёмкость h, то вместо тривиальной экспоненциальной оценки $\Delta^A(L) \leqslant 2^L$ верна более точная полиномиальная оценка

$$\Delta^A(L) \leqslant C_L^0 + C_L^1 + \cdots + C_L^h \leqslant \frac{3}{2}\frac{L^h}{h!}. \tag{3.11}$$

В этом случае правая часть (3.9) стремится к нулю при $\ell \to \infty$. Это означает, что семейство A обладает свойством *обучаемости* (learnability).

Теорема 3.4. *При $\ell = k$ для любого распределения на множестве $\mathbb{X}$ с вероятностью не менее $1 - \eta$ одновременно для всех алгоритмов $a \in A$ справедливо неравенство*

$$\nu(a, \bar{X}) < \nu(a, X) + \sqrt{\frac{h}{\ell} \ln\left(\frac{2e\ell}{h}\right) + \frac{4}{9\ell} \ln\frac{1}{\eta}}. \tag{3.12}$$

Доказательство. Подставим в (3.9) верхнюю оценку функции роста (4.1), оценив $h!$ снизу по формуле Стирлинга. Полученная оценка имеет вид $Q_\varepsilon \leqslant \eta(\varepsilon, \ell, h)$. Применим к ней технику обращения (1.8): выразим ε как функцию от ёмкости h, длины обучения ℓ и значения η. В результате получим (3.12). ■

Первое слагаемое в этой оценке — эмпирический риск. Он не возрастает с ростом ёмкости h, поскольку чем больше в семействе алгоритмов, тем более точно можно аппроксимировать выборку. Второе слагаемое возрастает с ростом ёмкости, и его можно рассматривать как *штраф за сложность* (complexity penalty). Сумма в общем случае достигает минимума при некотором h.

Для определения оптимальной сложности модели в VC-теории предлагается метод *структурной минимизации риска*. В семействе A заранее задаётся *структура* вложенных подсемейств возрастающей ёмкости $A_1 \subset A_2 \subset \cdots \subset A_h = A$. Задача обучения решается в каждом из этих подсемейств, всего h раз. Выбирается подсемейство оптимальной ёмкости, для которого достигается минимум правой части (3.12).

Метод структурной минимизации риска является важнейшим конструктивным следствием VC-теории. Проследим ещё раз общую логику всего, что было проделано. Сначала мы получили верхнюю оценку вероятности переобучения, которая справедлива для любой выборки $\mathbb{X}$ и любого метода обучения μ. К этой оценке мы применили технику обращения и получили верхнюю оценку для частоты ошибок на контроле. Она справедлива для любых $X, \bar{X}, \mu$, с вероятностью $(1 - \eta)$, близкой к единице. Теперь распорядимся свободой выбора μ так, чтобы минимизировать частоту ошибок на контроле. Полученный метод обучения называется *минимизацией оштрафованного эмпирического риска* (penalized empirical risk minimization).

Эта же общая логика эксплуатируется большинством современных подходов в теории статистического обучения [34].

§3.4 Проблема завышенности VC-оценок

Эксперимент 1: численные оценки требуемой длины обучения. Основная проблема VC-оценок в том, что они чрезвычайно завышены — настолько, что их применение практически теряет смысл. Чтобы в этом убедиться, достаточно выполнить численный расчёт требуемой длины обучающей выборки ℓ как функции от ёмкости h, точности ε и вероятности переобучения Q_ε.

Результаты приведены в Таблицах 3.1, 3.2, 3.3. Все данные вычислены при $\ell = k$. Первые две таблицы построены для общего случая, третья — для детерминистской постановки задачи. При построении первой таблицы использованы завышенные аппроксимации функции роста (4.1) и гипергеометрического сомножителя (3.8).

Основные выводы следующие.

1. Даже в детерминистском случае требуемая длина обучения на несколько порядков превышает те характерные длины выборок, с которыми обычно приходится иметь дело в прикладных задачах. Практика показывает, что что хорошая обучаемость возможна и по выборкам существенно меньшей длины.

	$\eta = 0.01$				$\eta = 1$			
h	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	60106	2404	601	150	14054	562	140	35
2	314692	9813	2149	460	265220	7786	1634	328
5	715120	21605	4631	961	665470	19565	4111	827
10	1386763	41427	8808	1806	1337061	39382	8287	1671
20	2733709	81218	17200	3504	2683987	79171	16677	3369
50	6780774	200844	42438	8616	6731042	198797	41916	8481
100	13530370	400406	84550	17149	13480635	398359	84027	17014

Таблица 3.1. Зависимость достаточной длины обучения ℓ от ёмкости h, точности ε и надёжности η, вычисленная согласно оценке $Q_\varepsilon \leqslant \eta = \frac{3}{2}\frac{L^h}{h!} \cdot \frac{3}{2}\exp(-\varepsilon^2\ell)$ из Теоремы 3.2. Это наименее точная оценка, использующая аппроксимацию функции роста (4.1) и аппроксимацию гипергеометрического сомножителя (3.8).

	$\eta = 0.01$				$\eta = 1$			
h	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	35900	1440	360	91	506	20	10	5
2	259300	7619	1600	316	210035	5579	1089	186
5	632633	18260	3770	741	582841	16219	3250	610
10	1262928	36396	7521	1470	1213200	34320	6989	1335
20	2531001	72918	15069	2936	2481120	70820	14549	2805
50	6348132	182980	37821	7381	6298001	180900	37290	7250
100	7373100	295440	73821	14811	7373100	295440	73821	14671

Таблица 3.2. Зависимость достаточной длины обучения ℓ от ёмкости h, точности ε и надёжности η, вычисленная согласно оценке $Q_\varepsilon \leqslant \eta = (C_L^0 + \cdots + C_L^h)\Gamma_L^\ell(\varepsilon, 1)$. Это также оценка Теоремы 3.2, но функция роста и гипергеометрический сомножитель вычисляются по точным формулам, без применения аппроксимаций.

	$\eta = 0.01$				$\eta = 1$			
h	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$	$\varepsilon = 0.01$	$\varepsilon = 0.05$	$\varepsilon = 0.1$	$\varepsilon = 0.2$
0	800	140	70	35	100	20	10	5
2	2800	440	200	85	2000	300	120	45
5	6200	960	410	165	5400	800	330	125
10	11900	1820	770	310	11200	1660	690	270
20	23500	3560	1500	600	22700	3400	1420	555
50	58100	8780	3700	1465	57400	8620	3620	1425
100	107000	17480	7370	2915	107000	17320	7290	2875

Таблица 3.3. Зависимость достаточной длины обучения от ёмкости h, точности ε и надёжности η, по детерминистской оценке $Q_\varepsilon \leqslant \eta = (C_L^0 + \cdots + C_L^h)C_{L-\lceil \varepsilon k \rceil}^\ell / C_L^\ell$ из Теоремы 3.3, без применений аппроксимаций.

2. Оценки в Таблице 3.1 существенно хуже, чем в таблицах 2 и 3. В дальнейшем мы будем избегать использования завышенных аппроксимаций и ставить задачу получения эффективных вычислительных методов, а не компактных формул[8].

3. Правые половины таблиц соответствуют значению $\eta = 1$ и показывают границу применимости VC-оценок. При меньших ℓ верхняя оценка вероятности вырождается — становится больше 1. Сопоставление правой и левой половин таблиц показывает, что достаточная длина обучения существенно зависит от ёмкости h и точности ε, но слабо зависит от требуемой вероятности переобучения η.

4. Первая строка таблицы соответствует семейству из одного алгоритма, $h = 0$. При этом достигается наилучшая возможная оценка. Однако этот случай не интересен с точки зрения статистического обучения.

5. Учёт априорной информации о корректности метода обучения уточняет VC-оценки, но, судя по доказательствам, не устраняет ни одного из основных факторов завышенности.

Причины завышенности VC-оценок видны из доказательства Теоремы 3.2, в котором сделаны три оценки сверху, а при получении Следствия 3.2.2 — ещё две. Классическая VC-оценка (3.9) не зависит от конкретной выборки $\mathbb{X}$ и метода обучения μ, следовательно, является оценкой худшего случая (worst case bound), который, скорее всего, никогда не реализуется на практике.

Эксперимент 2: оценки факторов завышенности. Чтобы понять, во сколько раз увеличивается оценка на каждом знаке $\leqslant$ в цепочке неравенств (3.6), был проведён эксперимент [59] с логическими закономерностями на шести реальных задачах из репозитория UCI, см. Таблицу 3.4. В каждой задаче методом Монте-Карло вычислялась оценка вероятности переобучения $\hat{Q}_\varepsilon$. Поделив её на функцию гипергеометрического распределения, можно получить *эффективный локальный коэффициент разнообразия* (ЭЛКР) $\hat{\Delta}$, показывающий, каким должно было бы быть значение Δ_L^ℓ, чтобы оценка вероятности переобучения не была завышенной. Поскольку логические закономерности строятся отдельно по классам, в таблице каждой задаче соответствует две строки (по числу классов). Опуская технические детали этого эксперимента, приведём лишь его результаты.

Под факторами завышенности понимаются коэффициенты r_1, r_2, r_3, вычисляемые по эмпирическим оценкам промежуточных членов в цепочке неравенств (3.6):

$$\hat{Q}_\varepsilon \cdot r_1 \cdot r_2 \cdot r_3 = \Delta^A(L) \cdot \tfrac{3}{2} e^{-\varepsilon\ell^2}.$$

Фактор r_1 возникает из-за того, что VC-оценки не учитывают эффекты локализации и расслоения семейства алгоритмов.

Эффект локализации состоит в том, что некоторые алгоритмы из A могут не выдаваться методом обучения μ ни на одной из обучающих подвыборок; в таком случае неравенство (3.7) окажется завышенным.

[8] В наши дни стремление получать простые асимптотические формулы является скорее пережитком докомпьютерной эпохи, чем насущной необходимостью. Мнение спорное, поэтому в сноске.

Таблица 3.4. Факторы завышенности r_1, r_2, r_3 и оценка $\hat{\Delta}$ с доверительным интервалом $[\hat{\Delta}_1; \hat{\Delta}_2]$.

Задача	класс	объектов	r_1	r_2	r_3	$\hat{\Delta}$	$[\hat{\Delta}_1; \hat{\Delta}_2]$
crx	0	307	2 759	680	32.6	24	[10; 41]
	1	383	1 104	1700	11.6	12	[11; 180]
german	1	300	15 215	1500	10.9	54	[38; 530]
	2	700	44 400	9000	9.9	1.9	[1.0; 2.2]
hepatitis	0	32	308	280	9.5	83	[11; 148]
	1	123	132	680	22.5	15	[12; 27]
horse-colic	1	191	151	4500	7.2	7	[2; 9]
	2	109	504	3400	7.3	6	[3; 6]
hypothyroid	0	3012	1 964 200	400	16.5	21	[3; 220]
	1	151	581 400	460	28.7	30	[2; 44]
promoters	0	53	555	340	9.8	72	[36; 230]
	1	53	510	790	6.9	18	[9; 22]

Эффект расслоения состоит в том, что на множестве алгоритмов $A_L^\ell(\mu, \mathbb{X})$ возникает существенно неравномерное распределение вероятности $P(a) = \mathsf{P}[\mu X = a]$. Алгоритмы с меньшим числом ошибок на генеральной выборке выдаются существенно чаще, в то же время, их существенно меньше. Эксперименты [44, 43] подтверждают, что профиль расслоения $\Delta(A_m, \mathbb{X})$, как правило, имеет форму узкого пика, сконцентрированного в средних слоях $m \approx L/2$.

Фактор r_2. VC-оценки не учитывают сходство алгоритмов в семействе. Применение неравенства Буля в (3.4) приводит к сильной завышенности, когда среди векторов ошибок имеется много похожих. Заметим, что *неравенство Буля*

$$\mathsf{P} \max_a [\delta(a, X) \geqslant \varepsilon] \leqslant \sum_a \mathsf{P}[\delta(a, X) \geqslant \varepsilon],$$

обращается в равенство только когда события $[\delta(a, X) \geqslant \varepsilon]$ несовместны. Если же векторы ошибок алгоритмов a, a' схожи, то соответствующие им события будут существенно совместными. На практике часто применяются *связные семейства* алгоритмов, в которых для каждого алгоритма $a \in A$ найдутся другие алгоритмы $a' \in A$ такие, что векторы ошибок $\vec{a}$ и $\vec{a}'$ отличаются только на одном объекте [53]. Связные семейства порождаются, в частности, методами классификации с непрерывной по параметрам разделяющей поверхностью. К ним относятся линейные классификаторы, машины опорных векторов с непрерывными ядрами, нейронные сети с непрерывными функциями активации, решающие деревья с пороговыми условиями ветвления, и многие другие. Таким образом, неравенство Буля сильнее всего завышено как раз в наиболее интересных с практической точки зрения случаях.

Фактор r_3 является «техническим» и связан, главным образом, с использованием аппроксимаций гипергеометрического распределения. Как видно из Таблицы 3.4, он наименее значим.

Заметим, что экспериментальные значения ЭЛКР имеют порядок $10^0 \dots 10^2$, тогда как функция роста находится в пределах $10^5 \dots 10^{10}$. Это означает, что VC-оценки завышены на несколько порядков. Более того, оценка ЭЛКР практически никогда не превышает длину выборки. Если бы мы захотели определить понятие *эффективной локальной ёмкости*, то она не превышала бы единицы, то есть была бы вырождена. Отсюда возникают сомнения в состоятельности и полезности понятия ёмкости для оценок переобучения.

В методе структурной минимизации риска завышенность оценок (3.2) и (3.12) приводит к выбору подсемейства заниженной ёмкости, то есть к переупрощению алгоритмов, что подтверждается и в экспериментах на модельных данных [42]. Распространённой ошибкой в интерпретации результатов VC-теории является вывод о необходимости ограничивать сложность семейства алгоритмов. Такой вывод был бы справедлив, если бы VC-оценки были достаточно точными. Пример алгоритма *бустинга* [38] показывает, что обобщающая способность может улучшаться даже при практически неограниченном росте сложности семейства.

О скользящем контроле. При практическом применении структурной минимизации риска вместо завышенной теоретической оценки (3.12) часто рекомендуют применять оценку скользящего контроля $\hat{\mathsf{E}}\nu(a, \bar{X})$. Однако такая замена ставит под сомнение ценность теоретических результатов, поскольку скользящий контроль не опирается на VC-теорию. С практической точки зрения скользящий контроль имеет ряд существенных недостатков. Во-первых, это ресурсоёмкая процедура. Во-вторых, оценка скользящего контроля имеет большую дисперсию, что может приводить к ошибкам при выборе оптимальной сложности подсемейства h. В-третьих, скользящий контроль удобен для эмпирического оценивания качества метода обучения, но не удобен для конструирования новых методов. По этим причинам задача получения точных теоретических оценок не теряет актуальности.

Резюме

VC-теория основана на принципе равномерной сходимости. Она позволяет оценивать вероятность переобучения сверху функцией от длины выборки и сложности семейства алгоритмов. Мерой сложности является коэффициент разнообразия, определяемый как число попарно различных бинарных векторов ошибок, индуцируемых всевозможными алгоритмами семейства на заданной выборке.

Конструктивным выходом VC-теории является метод структурной минимизации риска. Он приводит к принципу минимизации оштрафованного эмпирического риска. Вследствие завышенности оценок его непосредственное применение может приводить к неоптимальному выбору слишком простых алгоритмов.

Основной причиной завышенности VC-оценок является пренебрежение эффектами расслоения и связности, которыми обладают многие семейства алгоритмов, применяемые на практике.

В следующей лекции будут рассмотрены верхние оценки функции роста и ёмкости как для общего случая, так и для некоторых конкретных семейств алгоритмов. Казалось бы, это некоторое отступление от основной цели курса — получения точных оценок вероятности переобучения. Тем не менее, оценивание функции роста — это классическая и весьма полезная комбинаторная техника, необходимая для понимания внутренней структуры семейств алгоритмов.

Упражнения

Задача 3.1 (1). Доказать Теорему 3.4, используя формулу Стирлинга.

Задача 3.2 (2). *Степенью некорректности* метода обучения μ на выборке $\mathbb{X}$ называется максимальная частота ошибок на всевозможных обучающих подвыборках: $\sigma(\mu, \mathbb{X}) = \max\limits_{X \in [\mathbb{X}]^\ell} \nu(\mu X, X)$. Доказать, что для любых μ, $\mathbb{X}$ с ограниченной некорректностью $\sigma(\mu, \mathbb{X}) \leqslant \sigma$ и любого $\varepsilon \in [0, 1]$ справедлива оценка, обобщающая (3.2) и (3.10):

$$Q_\varepsilon \leqslant \Delta_L^\ell \max_{m \in M(\varepsilon,\sigma)} H_L^{\ell,m}\left(s_m(\varepsilon, \sigma)\right), \tag{3.13}$$

где $M(\varepsilon, \sigma) = \{m \colon \varepsilon k \leqslant m \leqslant k + \sigma\ell\}$, $s_m(\varepsilon, \sigma) = \min\bigl\{s_m(\varepsilon), \sigma\ell\bigr\}$.

Построить графики зависимости отношения оценки (3.2) к оценке (3.13) от степени некорректности σ.

Задача 3.3 (2). Следуя [33], показать, что если множество векторов ошибок $\bigl\{(a)_{\mathbb{X}} \colon a \in A\bigr\}$ кластеризуется по расстоянию Хэмминга на $S(r)$ кластеров радиуса r каждый, то

$$\mathsf{P}\bigl[\delta_\mu(X) \geqslant \varepsilon + \tfrac{r}{\ell}\bigr] \leqslant S(r) \cdot \max_{m=1,\ldots,L} H_L^{\ell,m}\left(s_1(\varepsilon)\right).$$

4 Размерность Вапника-Червоненкиса

В теории Вапника-Червоненкиса доказывается, что функция роста $\Delta^A(L)$ либо равна 2^L, либо растёт полиномиально по L, причём промежуточных вариантов не существует. Как следует из (**??**), в полиномиальном случае правая часть оценки стремится к нулю при $\ell, k \to \infty$, следовательно, обучение асимптотически состоятельно. Разберём этот фундаментальный результат более подробно.

§4.1 Определение ёмкости и её связь с функцией роста

Определение 4.1. *Если существует целое число h такое, что $\Delta^A(h) = 2^h$ и $\Delta^A(h+1) < 2^{h+1}$, то оно называется ёмкостью или размерностью Вапника-Червоненкиса (VC-dimension) семейства алгоритмов A. Если такого числа h не существует, то говорят, что семейство A имеет бесконечную ёмкость.*

Если семейство имеет конечную ёмкость h, то его функцию роста можно оценить сверху величиной, зависящей от L полиномиально при $L > h$. Для доказательства этого факта нам понадобятся некоторые вспомогательные построения.

Лемма 4.1. *Функция $\Phi_L^h = C_L^0 + C_L^1 + \cdots + C_L^h$, определённая при целых h и L, таких, что $0 \leqslant h \leqslant L$, однозначно задаётся рекуррентными соотношениями*

$$\Phi_L^0 = 1, \qquad \Phi_L^L = 2^L, \qquad \Phi_L^h = \Phi_{L-1}^h + \Phi_{L-1}^{h-1}, \qquad 0 \leqslant h \leqslant L.$$

Доказательство следует из того, что биномиальные коэффициенты C_L^h определяются аналогичным рекуррентным соотношением $C_L^h = C_{L-1}^h + C_{L-1}^{h-1}$ и отличаются только граничным условием $C_L^L = 1$.

Лемма 4.2 (Вапник, Червоненкис [6]). *Если для любой подвыборки X^{h+1} из X^L выполняется $\Delta^A(X^{h+1}) < 2^{h+1}$, то $\Delta^A(X^L) \leqslant \Phi_L^h$.*

Доказательство. Доказательство проведём индукцией по h.

При $h = 0$ из того, что $\Delta^A(x_i) < 2$ для всех $x_i \in X^L$ вытекает $\Delta^A(X^L) = 1 = \Phi_L^0$, следовательно, утверждение леммы справедливо. Предполагая, что оно справедливо для $h-1$, покажем, что оно справедливо также и для h при всех L, больших h.

Для этого при фиксированном h применим индукцию по L.

При $L = h+1$ имеем $\Delta^A(X^{h+1}) \leqslant 2^{h+1} - 1 = \Phi_{h+1}^h$, значит утверждение леммы выполнено. Допустим теперь, что оно выполняется для $\Delta^A(X^L)$, и оценим сверху $\Delta^A(X^{L+1})$. Представим выборку X^{L+1} в виде (X^L, x_{L+1}).

Будем говорить, что алгоритм a на заданной выборке U *индуцирует подвыборку* U', если $U' = \{x \in U \colon a(x) = 1\}$. Рассмотрим множество всех подвыборок, индуцируемых на X^L всеми алгоритмами семейства A. Будем различать подвыборки двух типов:

1) такие подвыборки X^r из X^L, что алгоритмы семейства A индуцируют на X^{L+1} как X^r, так и (X^r, x_{L+1});

2) все остальные подвыборки.

Обозначим число подвыборок первого типа K_1, а второго типа K_2. Тогда

$$\Delta^A(X^L) = K_1 + K_2,$$
$$\Delta^A(X^{L+1}) = 2K_1 + K_2,$$

следовательно, $\Delta^A(X^{L+1}) = \Delta^A(X^L) + K_1$.

Рассмотрим подмножество алгоритмов A', индуцирующих на X^L только подвыборки первого типа. Тогда $K_1 = \Delta^{A'}(X^L)$.

Имеется две возможности.

1. Допустим, найдётся подвыборка $X^h \subseteq X^L$ такая, что $\Delta^{A'}(X^h) = 2^h$. Это означает, что алгоритмы множества A' индуцируют на X^h, а значит и на (X^h, x_{L+1}), все возможные подвыборки $X^r \subseteq X^h$. По определению множества A' на (X^h, x_{L+1}) индуцируются также все подвыборки вида (X^r, x_{L+1}). Следовательно

$$\Delta^{A'}(X^h, x_{L+1}) = 2^h + 2^h = 2^{h+1}.$$

Но тогда $\Delta^A(X^h, x_{L+1}) = 2^{h+1}$, что противоречит условию леммы.

2. Допустим теперь, что для любой подвыборки $X^h \subseteq X^L$ выполнено условие $\Delta^{A'}(X^h) < 2^h$. По предположению индукции отсюда вытекает $\Delta^{A'}(X^L) \leqslant \Phi_L^{h-1}$. Таким образом

$$\Delta^A(X^{L+1}) = \Delta^A(X^L) + \Delta^{A'}(X^L) \leqslant \Phi_L^h + \Phi_L^{h-1} = \Phi_{L+1}^h$$

Утверждение индукции доказано для $L + 1$. ■

Лемма 4.3. *Справедлива оценка* $\Phi_L^h \leqslant 1.5\frac{L^h}{h!}, \quad 0 \leqslant h \leqslant L.$

Доказательство является несложным техническим упражнением [6].

Теорема 4.4 (Вапник, Червоненкис [6]). *Если семейство A имеет конечную ёмкость h, то при $L > h$ функция роста $\Delta^A(L)$ зависит от L полиномиально:*

$$\Delta^A(L) \leqslant \Phi_L^h \leqslant 1.5\frac{L^h}{h!}. \tag{4.1}$$

Доказательство. Пусть $L \leqslant h$. Тогда из условия $\Delta^A(h) = 2^h$ вытекает, что существует выборка длины L, на которой алгоритмы семейства A индуцируют все возможные подвыборки. Значит $\Delta^A(L) = 2^L$.

Пусть $L \geqslant h$. Возьмём произвольную выборку X^L. Для неё выполнено условие леммы 4.2, так как $\Delta^A(h+1) < 2^{h+1}$. Следовательно $\Delta^A(X^L) \leqslant \Phi_L^h$, и в силу произвольности выборки $\Delta^A(L) \leqslant \Phi_L^h$.

Теорема доказана. ■

§4.2 Функция роста и ёмкость конечного множества

Пусть множество A конечно. Число алгоритмов, попарно неразличимых на выборке X^L, не превышает числа всех алгоритмов, поэтому для функции роста справедлива оценка

$$\Delta^A(L) \leqslant |A|.$$

Ёмкость такого семейства не превышает $\lfloor \log_2 |A| \rfloor$, так как в противном случае функция роста оказалась бы больше $|A|$.

Множества алгоритмов, реализуемых на компьютере, всегда конечны. Если для хранения всех параметров алгоритма используется не более n бит, то число алгоритмов в таком семействе не превышает 2^n, а его ёмкость не превышает $\log_2 2^n = n$. Чтобы эта оценка не была завышенной, для подсчёта необходимого числа бит должно использоваться *максимально экономное кодирование* параметров [16].

Согласно теореме 3.2, чем меньше длина записи алгоритма, тем точнее оценивается частота ошибок на контроле по частоте ошибок на обучении. Отсюда вытекает т. н. принцип *минимума длины описания* (Minimal Description Length, MDL) [51].

§4.3 Функция роста множества конъюнкций

Для случая, когда объекты описываются дискретными признаками $f_j\colon X \to D_j$, $|D_j| < \infty$, оценим функцию роста множества всех конъюнкций ранга не выше K:

$$A = \Big\{ a(x) = \bigwedge_{j\in J} [f_j(x) = d_j] \;\Big|\; J \subseteq \{1,\ldots,n\},\ |J| \leqslant K,\ d_j \in D_j \Big\}.$$

Если J — произвольное подмножество индексов из $\{1,\ldots,n\}$, то число конъюнкций ранга k, которые можно построить по признакам из J, есть

$$H_k(J) = \sum_{\substack{J'\subseteq J \\ |J'|=k}} \prod_{j\in J'} |D_j|.$$

Если множества D_j равномощны, $|D_j| = d$, то $H_k(J) = C^k_{|J|} d^k$. В общем случае величина $H_k(J)$ легко вычисляется по рекуррентным соотношениям:

$$H_0(J) = 1;$$
$$H_k(J) = 0, \quad k > |J|;$$
$$H_k\big(J \cup \{j\}\big) = H_k(J) + |D_j|\, H_{k-1}(J), \quad k < |J|, \quad j = 1,\ldots,n.$$

Функция роста оценивается сверху числом конъюнкций ранга не выше K, которые можно построить по всем n признакам:

$$\Delta^A(L) \leqslant \sum_{k=1}^{K} H_k\{1,\ldots,n\}.$$

§4.4 Ёмкость семейства линейных классификаторов

Пусть $X = \mathbb{R}^n$, $Y = \{0, 1\}$, A — семейство линейных классификаторов:

$$A = \big\{a(x) = [\langle w, x\rangle \geqslant 0] \;\big|\; w \in \mathbb{R}^n\big\},$$

где $\langle w, x\rangle$ — скалярное произведение векторов w и x. Каждый алгоритм этого семейства задаётся вектором w из $\mathbb{R}^n$.

Теорема 4.5. *Ёмкость семейства линейных классификаторов A равна размерности пространства n.*

Идея доказательства заключается в том, что в пространстве размерности n через произвольные n точек можно провести разделяющую гиперплоскость, а через некоторые $n + 1$ — уже нельзя.

Доказательство. Покажем сначала, что $\Delta^A(n) = 2^n$. Согласно определению функции роста это равносильно следующему высказыванию:

$$\exists X^n \quad \forall (z_1, \ldots, z_n) \in Y^n \quad \exists a \in A \quad \forall i = 1, \ldots, n \quad a(x_i) = z_i.$$

Возьмём n векторов $X^n = \{x_1, \ldots, x_n\}$ из X таких, что у i-ого вектора i-ая компонента равна 1, а остальные равны 0. Рассмотрим алгоритм $a \in A$, задаваемый вектором коэффициентов $w = (w_1, \ldots, w_n)$. Каким бы ни был бинарный вектор $(z_1, \ldots, z_n)$, легко подобрать коэффициенты w_i так, чтобы выполнялось $a(x_i) = [w_i \geqslant 0] = z_i$. Таким образом, мы указали 2^n алгоритмов, различным образом делящих выборку X^n на два класса.

Теперь покажем, что $\Delta^A(n + 1) < 2^{n+1}$. Это равносильно высказыванию

$$\forall X^{n+1} \quad \exists (z_1, \ldots, z_{n+1}) \in Y^{n+1} \quad \forall a \in A \quad \exists i = 1, \ldots, n + 1 \quad a(x_i) \neq z_i.$$

Возьмём произвольные $n + 1$ векторов $x_1, \ldots, x_{n+1}$ из X. Число векторов превышает их размерность, поэтому среди них найдётся хотя бы один, являющийся линейной комбинацией остальных. Допустим без ограничения общности, что это x_{n+1}:

$$x_{n+1} = b_1 x_1 + \cdots + b_n x_n, \tag{4.2}$$

где $b_1, \ldots, b_n$ — действительные числа.

Положим $z_i = [b_i \geqslant 0]$ для всех $i = 1, \ldots, n$ и $z_{n+1} = 0$. Рассмотрим произвольный алгоритм $a \in A$ с коэффициентами $w = (w_1, \ldots, w_n)$. Допустим, что $a(x_i) = z_i$ для всех $i = 1, \ldots, n + 1$. Умножим обе части равенства (4.2) скалярно на w:

$$\langle w, x_{n+1}\rangle = b_1 \langle w, x_1\rangle + \cdots + b_n \langle w, x_n\rangle .$$

Левая часть этого равенства строго меньше нуля, поскольку

$$[\langle w, x_{n+1}\rangle \geqslant 0] = a(x_{n+1}) = z_{n+1} = 0.$$

В то же время, каждое слагаемое в правой части равенства неотрицательно, так как

$$[\langle w, x_i\rangle \geqslant 0] = a(x_i) = z_i = [b_i \geqslant 0], \quad i = 1, \ldots, n.$$

Таким образом, сделанное допущение приводит к противоречию. Какой бы ни была выборка X^{n+1}, алгоритмы из A не реализуют всех 2^{n+1} способов поделить её на 2 класса.

Теорема доказана. ■

§4.5 Однопараметрическое семейство бесконечной ёмкости

что свидетельствует о нетривиальности понятия ёмкости с одной стороны, и о бесконечных выразительных способностях действительного числа с другой [56].

Рассмотрим семейство функций $a\colon \mathbb{R} \to \{0, 1\}$ с одним параметром $\gamma \in \mathbb{R}$:

$$a(x; \gamma) = \big[\sin(\gamma x) < 0\big].$$

Возьмём конкретную выборку объектов $x_i = 10^{-i}$, $i = 1, \ldots, \ell$. Какова бы ни была её длина ℓ, для любого вектора ответов $(y_i)_{i=1}^{\ell}$ можно так подобрать параметр γ, чтобы $a(x_i; \gamma) = y_i$. Действительно, возьмём $\gamma = \pi + \pi \sum_{j=1}^{\ell} y_j 10^j$. Тогда

$$a(x_i; \gamma) = \left[\sin\Big(\pi y_i + \underbrace{\pi 10^{-i} + \pi \sum\nolimits_{j=1}^{i-1} y_j 10^{i-j}}_{\gamma_0}\Big) < 0\right] = \begin{cases} [\sin\gamma_0 < 0], & y_i = 0; \\ [\sin\gamma_0 > 0], & y_i = 1. \end{cases}$$

Из определения величины γ_0 следует, что $\pi 10^{-\ell} \leqslant \gamma_0 \leqslant 0.3\pi$, поэтому значение $\sin\gamma_0$ положительно, и правая часть равенства есть просто y_i.

Рассмотренный пример является искусственным. Если для представления числа γ использовать конечное число бит, ёмкость уже не будет бесконечной.

§4.6 Другие оценки ёмкости

Оценки ёмкости были получены для нейронных сетей [31, 30, 41, 48], решающих деревьев [17], корректных полиномов над алгоритмами вычисления оценок [25], комитетных решающих правил [47], и других семейств.

Ёмкость — нетривиальное понятие, и далеко не всегда она связана с числом параметров алгоритма. Известны примеры многопараметрических семейств ёмкости 1 и, как мы уже видели, однопараметрических семейств бесконечной ёмкости.

Ёмкость семейств, основанных на явном хранении всей обучающей выборки, как правило, бесконечна (например, у алгоритма ближайших соседей). Ёмкость семейств, гарантирующих корректность (отсутствие ошибок) на обучающей выборке, также, как правило, бесконечна. Хотя, есть и исключения: в работах В. Л. Матросова строятся композиции алгоритмов вычисления оценок, имеющие конечную ёмкость и одновременно гарантирующие корректность [24, 25, 26, 27].

Резюме

Ёмкость или размерность Вапника-Червоненкиса — это классическая характеристика сложности семейств алгоритмов. Название оправдывается тем, что для линейных семейств это буквально размерность пространства параметров. Однако для других семейств ёмкость не столь тривиально выражается через число параметров. Существуют примеры, когда параметров много, а ёмкость равна единице, и, наоборот, когда параметр один, а ёмкость бесконечна.

В следующей лекции мы убедимся, что знания одного лишь коэффициента разнообразия категорически недостаточно для получения точных оценок вероятности переобучения. Не достаточно знать, сколько попарно различных векторов ошибок семейство алгоритмов индуцирует на данной выборке. Необходимо ещё учитывать степень их различности.

5 Порождающие и запрещающие множества

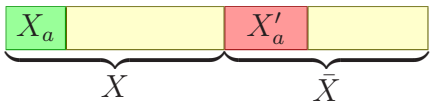
Принцип порождающих и запрещающих множеств (ПЗМ) позволяет получать точные (не завышенные, не асимптотические) оценки обобщающей способности [62]. Он основан на предположении, что для каждого алгоритма можно выписать необходимые и достаточные условия того, что он является результатом обучения. Если же удаётся выписать лишь необходимые условия, то получаются верхние оценки.

§5.1 Простая гипотеза ПЗМ

Гипотеза 5.1. *Пусть множество A, выборка $\mathbb{X}$ и метод μ таковы, что для каждого алгоритма $a \in A$ можно указать пару непересекающихся подмножеств $X_a \subset \mathbb{X}$ и $X'_a \subset \mathbb{X}$, удовлетворяющую условию*

$$\big[\mu X{=}a\big] = \big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big], \quad \forall X \in [\mathbb{X}]^\ell. \tag{5.1}$$

Множество X_a будем называть *порождающим*, множество X'_a — *запрещающим* для алгоритма a. Гипотеза 5.1 означает, что метод μ выбирает алгоритм a тогда и только тогда, когда в обучающей выборке X находятся все порождающие объекты и ни одного запрещающего:

$\mu X = a$: 

Все остальные объекты $\mathbb{X}\backslash X_a\backslash X'_a$ будем называть *нейтральными* для алгоритма a. Наличие или отсутствие нейтральных объектов в обучающей выборке не влияет на результат обучения. Далее будут приведены примеры семейств, для которых гипотеза 5.1 выполняется.

Лемма 5.1. *Для любой выборки X справедливо тождество*

$$\sum_{a\in A}\big[\mu X{=}a\big] = 1. \tag{5.2}$$

Доказательство с очевидностью вытекает из того, что для любой выборки X метод μ выбирает один и только один алгоритм. Тождество (5.2) может использоваться для проверки того, что условия в правой части (5.1) сформулированы корректно.

Для произвольного $a \in A$ обозначим через L_a число нейтральных объектов, через ℓ_a — число нейтральных объектов, попадающих в обучающую выборку:

$$L_a = L - |X_a| - |X'_a|;$$
$$\ell_a = \ell - |X_a|.$$

Лемма 5.2. *Если гипотеза 5.1 справедлива, то вероятность получить в результате обучения алгоритм a равна*

$$P_a = \mathsf{P}\big[\mu X{=}a\big] = \frac{C^{\ell_a}_{L_a}}{C^{\ell}_{L}}.$$

Доказательство. Согласно гипотезе 5.1

$$\mathsf{P}\big[\mu X{=}a\big] = \mathsf{P}\big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big].$$

Это есть доля разбиений генеральной выборки $\mathbb{X} = X \sqcup \bar{X}$ таких, что множество объектов X_a целиком лежит в X, а множество объектов X'_a целиком лежит в $\bar{X}$. Число таких разбиений равно числу способов отобрать ℓ_a из L_a нейтральных объектов в обучающую подвыборку $X\backslash X_a$, которое, очевидно, равно $C_{L_a}^{\ell_a}$. Общее число разбиений равно C_L^ℓ, а их отношение как раз и есть P_a. ■

Вероятность переобучения Q_ε выражается по формуле полной вероятности, если для каждого алгоритма a из A известна вероятность P_a получить его в результате обучения и условная вероятность большого отклонения частот $\mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \mid a\big)$ при условии, что получен алгоритм a:

$$Q_\varepsilon = \sum_{a\in A} P_a\, \mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \;\big|\; a\big).$$

Условная вероятность даётся Теоремой 2.4, если учесть, что при фиксированном алгоритме a подмножества X_a и X'_a не участвуют в разбиениях. Рассматривая L_a нейтральных объектов и всевозможные их разбиения на ℓ_a обучающих и $L_a - \ell_a$ контрольных, получим:

$$\mathsf{P}\big(\delta(a,X) \geqslant \varepsilon \;\big|\; a\big) = H_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right),$$

где m_a — число ошибок алгоритма a на нейтральных объектах; $s_a(\varepsilon)$ — наибольшее число ошибок алгоритма a на нейтральных обучающих объектах $X\setminus X_a$, при котором имеет место большое отклонение частот ошибок, $\delta(a,X) \geqslant \varepsilon$:

$$\begin{aligned} m_a &= n(a, \mathbb{X}\backslash X_a\backslash X'_a); \\ s_a(\varepsilon) &= \tfrac{\ell}{L}\big(n(a,\mathbb{X}) - \varepsilon k\big) - n(a,X_a). \end{aligned}$$

Пока это было не доказательство, а лишь наводящие соображения. Трюк с условной вероятностью может показаться не вполне очевидным. Ниже представлен строгий комбинаторный вывод точной оценки Q_ε.

Теорема 5.3. *Если гипотеза 5.1 справедлива, то вероятность переобучения вычисляется по формуле*

$$Q_\varepsilon(\mu,\mathbb{X}) = \sum_{a\in A} P_a H_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right).$$

Доказательство. Рассмотрим функционал Q_ε. Введём в (1.5) под знак суммирования по X ещё два вспомогательных суммирования: первый — по всем алгоритмам a из A при условии $\mu X = a$, второй — по всем значениям s числа ошибок алгоритма a на подвыборке $X\backslash X_a$. Очевидно, значение Q_ε от этого не изменится:

$$Q_\varepsilon = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\sum_{a\in A}\big[\mu X{=}a\big]\sum_{s=0}^{\ell_a}\big[n(a,X\backslash X_a) = s\big]\big[\delta(a,X) \geqslant \varepsilon\big]. \tag{5.3}$$

Число ошибок алгоритма a на обучающей подвыборке X равно $s + n(a, X_a)$, поэтому отклонение частот ошибок выражается в виде

$$\delta(a, X) = \frac{n(a, \mathbb{X}) - s - n(a, X_a)}{k} - \frac{s + n(a, X_a)}{\ell},$$

следовательно,

$$\big[\delta(a, X) \geqslant \varepsilon\big] = \big[s \leqslant \tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_a)\big] = \big[s \leqslant s_a(\varepsilon)\big].$$

Подставим полученное выражение в (5.3), затем заменим $\big[\mu X = a\big]$ правой частью равенства (5.1) и переставим знаки суммирования (очевидно, P также можно рассматривать как суммирование):

$$Q_\varepsilon = \sum_{a \in A} \sum_{s=0}^{\ell_a} \underbrace{\mathsf{P}\big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big]\big[n(a, X \backslash X_a) = s\big]}_{N(a)} \big[s \leqslant s_a(\varepsilon)\big]. \tag{5.4}$$

Выделенное в данной формуле выражение $N(a)$ есть доля разбиений генеральной выборки $\mathbb{X} = X \sqcup \bar{X}$ таких, что множество объектов X_a целиком лежит в X, множество объектов X'_a целиком лежит в $\bar{X}$, и в подвыборку $X \backslash X_a$ длины ℓ_a попадает ровно s объектов, на которых алгоритм a допускает ошибку.

Для наглядности представим вектор ошибок a разбитым на шесть блоков:

$$\vec{a} = \Big(\underbrace{X_a; \overbrace{1, \ldots, 1}^{s}; 0, \ldots, 0}_{X \backslash X_a}; \underbrace{X'_a; \overbrace{1, \ldots, 1}^{m_a - s}; 0, \ldots, 0}_{\bar{X} \backslash X'_a} \Big).$$

Число ошибок алгоритма a на объектах, не попадающих ни в X_a, ни в X'_a, равно m_a. Существует $C^s_{m_a}$ способов выбрать из них s объектов, которые попадут в $X \backslash X_a$. Для каждого из этих способов имеется ровно $C^{\ell_a - s}_{L_a - m_a}$ способов выбрать $\ell_a - s$ объектов, на которых алгоритм a не допускает ошибку, и которые также попадут в $X \backslash X_a$. Тем самым однозначно определяется состав выборки $X \backslash X_a$, а, значит, и состав выборки $\bar{X} \backslash X'_a$. Таким образом, $N(a) = C^s_{m_a} C^{\ell_a - s}_{L_a - m_a} / C^\ell_L$. Подставим это выражение в (5.4) и выделим в нём формулу гипергеометрической функции вероятности:

$$Q_\varepsilon = \sum_{a \in A} \frac{C^{\ell_a}_{L_a}}{C^\ell_L} \sum_{s=s_0}^{\ell_a} \big[s \leqslant s_a(\varepsilon)\big] \frac{C^s_{m_a} C^{\ell_a - s}_{L_a - m_a}}{C^{\ell_a}_{L_a}} = \sum_{a \in A} P_a H^{\ell_a,\, m_a}_{L_a}\big(s_a(\varepsilon)\big).$$

Теорема доказана. ■

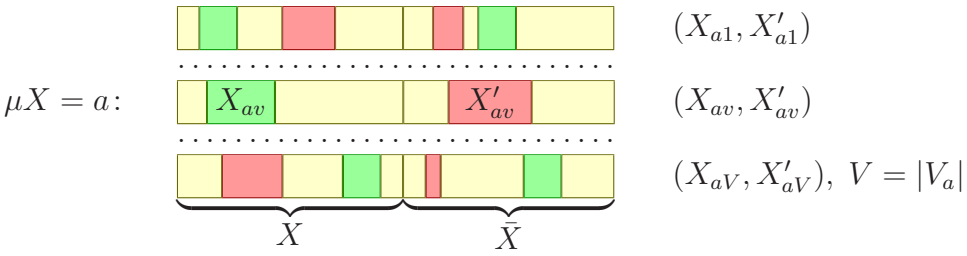
§5.2 Обобщённая гипотеза ПЗМ

Гипотеза 5.1 накладывает слишком сильные ограничения на выборку $\mathbb{X}$, семейство A и метод μ. Поэтому Теорему 5.3 удаётся применять лишь в некоторых специальных случаях. Рассмотрим естественное обобщение гипотезы 5.1. Предположим, что для каждого алгоритма a существуют различные варианты выделения порождающих и запрещающих множеств.

Гипотеза 5.2. *Пусть множество A, выборка $\mathbb{X}$ и метод μ таковы, что для каждого алгоритма $a \in A$ можно указать конечное множество индексов V_a, и для каждого индекса $v \in V_a$ можно указать порождающее множество $X_{av} \subset \mathbb{X}$, запрещающее множество $X'_{av} \subset \mathbb{X}$ и коэффициент $c_{av} \in \mathbb{R}$, удовлетворяющие условиям*

$$\big[\mu X{=}a\big] = \sum_{v \in V_a} c_{av} \big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big], \quad \forall X \in [\mathbb{X}]^\ell. \tag{5.5}$$

При условии $c_{av} \equiv 1$ гипотеза 5.2 означает, что метод μ выбирает алгоритм a тогда и только тогда, когда в обучающую выборку X попадают все объекты из X_{av} и ни одного из X'_{av}, ровно для одной из пар множеств (X_{av}, X'_{av}), $v \in V_a$:



Очевидно, условие (5.5) должно быть задано так, чтобы правая часть принимала только два значения — либо 0, либо 1. Это требование накладывает определённые ограничения и на систему подмножеств (X_{av}, X'_{av}), $v \in V_a$, и на коэффициенты c_{av}. В частности, при $c_{av} \equiv 1$ условия $\big[X_{av} \subseteq X\big]\big[X'_{av} \subseteq \bar{X}\big]$ не могут выполняться одновременно для двух индексов $v, v' \in V_a$ ни при каком разбиении $(X, \bar{X})$, иначе значение в правой части окажется большим единицы.

К настоящему моменту не известны случаи, когда приходилось бы задавать коэффициенты c_{av}, отличные от $+1$ или -1.

Очевидно, гипотеза 5.1 является частным случаем гипотезы 5.2, когда все множества V_a одноэлементные и $c_{av} = 1$.

Следующая теорема утверждает, что гипотеза 5.2 верна всегда.

Теорема 5.4. *Для любых $\mathbb{X}$, A и μ существуют множества V_a, X_{av}, X'_{av}, при которых справедливо представление* (5.5)*, причём $c_{av} = 1$ для всех $a \in A$, $v \in V_a$.*

Доказательство. Зафиксируем произвольный алгоритм $a \in A$. Возьмём в качестве индексного множества V_a множество всех подвыборок $v \in [\mathbb{X}]^\ell$, при которых $\mu v = a$. Для каждого $v \in V_a$ положим $X_{av} = v$, $X'_{av} = \mathbb{X} \setminus v$, $c_{av} = 1$. Тогда для любого $X \in [\mathbb{X}]^\ell$ справедливо представление, имеющее вид (5.5):

$$\big[\mu X{=}a\big] = \sum_{v \in V_a} \big[v{=}X\big] = \sum_{v \in V_a} \big[v{=}X\big] \big[\mathbb{X}\backslash v{=}\mathbb{X}\backslash X\big] = \sum_{v \in V_a} \big[v \subseteq X\big] \big[\mathbb{X}\backslash v \subseteq \bar{X}\big],$$

причём, если $\mu X = a$, то ровно одно слагаемое в этой сумме равно единице, остальные равны нулю; если же $\mu X \neq a$, то все слагаемые равны нулю. ■

Теорема 5.4 является типичной теоремой существования. Использованный при её доказательстве способ построения индексных множеств V_a требует явного перебора всех разбиений выборки, что приводит к вычислительно неэффективным оценкам

вероятности переобучения. Однако представление (5.5) в общем случае не единственно. Отдельной проблемой является поиск такого представления, в котором мощности множеств $|V_a|$, $|X_{av}|$, $|X'_{av}|$ были бы как можно меньше. Хотя гипотеза 5.2 верна всегда, мы будем продолжать называть её «гипотезой», имея в виду предположение о существовании некоторого представления вида (5.5), более эффективного, чем использованное в доказательстве Теоремы 5.4.

Введём для каждого алгоритма $a \in A$ и каждого индекса $v \in V_a$ обозначения:

$$\begin{aligned} L_{av} &= L - |X_{av}| - |X'_{av}|; \\ \ell_{av} &= \ell - |X_{av}|; \\ m_{av} &= n(a, \mathbb{X}) - n(a, X_{av}) - n(a, X'_{av}); \\ s_{av}(\varepsilon) &= \tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_{av}). \end{aligned}$$

В условиях гипотезы 5.2 справедливы соответствующие обобщения леммы о вероятностях получения алгоритмов и теоремы о вероятности переобучения.

Лемма 5.5. *Если гипотеза 5.2 справедлива, то для всех $a \in A$ вероятность получить в результате обучения алгоритм a равна*

$$P_a = \mathsf{P}\big[\mu X{=}a\big] = \sum_{v \in V_a} c_{av} P_{av}; \tag{5.6}$$

$$P_{av} = \mathsf{P}\big[X_{av} \subseteq X\big]\big[X'_{av} \subseteq \bar{X}\big] = \frac{C^{\ell_{av}}_{L_{av}}}{C^{\ell}_{L}}. \tag{5.7}$$

Доказательство. Достаточно применить операцию P к левой и правой частям (5.5). Дальнейшие рассуждения аналогичны доказательству Леммы 5.2. ■

Теорема 5.6. *Если гипотеза 5.2 справедлива, то вероятность переобучения вычисляется по формуле*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} \sum_{v \in V_a} c_{av} P_{av} H^{\ell_{av},\, m_{av}}_{L_{av}}\left(s_{av}(\varepsilon)\right). \tag{5.8}$$

Доказательство. Аналогично доказательству Теоремы 5.3, вероятность переобучения приводится к выражению, которое отличается от (5.4) появлением знака суммирования по v, коэффициентов c_{av} и двойных индексов $_{av}$ вместо одинарных $_a$:

$$Q_\varepsilon = \sum_{a \in A} \sum_{v \in V_a} \sum_{s=0}^{\ell} c_{av}\, \mathsf{P}\big[X_{av} \subseteq X\big]\big[X'_{av} \subseteq \bar{X}\big]\big[n(a, X \backslash X_{av}) = s\big]\big[s \leqslant s_{av}(\varepsilon)\big],$$

В остальном доказательство аналогично доказательству Теоремы 5.3. ■

Оценки функционала R_ε (1.6) доказываются аналогично, с той лишь разницей, что выражение

$$s_{av}(\varepsilon) = \tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_{av})$$

всюду заменяется на

$$s'_{av}(\varepsilon) = n(a, \mathbb{X}) - \varepsilon k - n(a, X_{av}).$$

§5.3 Корректное семейство алгоритмов

Алгоритм a, не допускающий ошибок на выборке $U \subseteq \mathbb{X}$, называется *корректным на выборке* U. Если множество A содержит алгоритм a_0, корректный на генеральной выборке $\mathbb{X}$, то множество A будем называть *корректным*. В этом случае формула (5.8) сильно упрощается.

Лемма 5.7. *В случае $m = 0$ функция гипергеометрического распределения вырождается:* $H_L^{\ell,0}(s) = [s \geqslant 0]$.

Доказательство. При $s \geqslant 0$ сумма $H_L^{\ell,0}(s)$ в (2.4) состоит из одного слагаемого, равного 1. При $s < 0$ число слагаемых равно нулю, и вся сумма равна нулю. ■

Теорема 5.8. *Пусть гипотеза 5.2 справедлива, метод μ является минимизацией эмпирического риска, множество A корректно. Тогда вероятность переобучения принимает более простой вид:*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{a \in A} \big[n(a, \mathbb{X}) \geqslant \varepsilon k\big] P_a. \tag{5.9}$$

Доказательство. Рассмотрим произвольный алгоритм $a \in A$ и произвольный индекс $v \in V_a$. Если некоторый объект, на котором a допускает ошибку, содержится в обучающей выборке X, то метод μ, минимизирующий эмпирический риск, не сможет выбрать данный алгоритм, так как существует корректный алгоритм a_0, не допускающий ошибок на X. Следовательно, множество объектов, на которых алгоритм a допускает ошибку, целиком содержится в X'_{av}. Значит, алгоритм a не допускает ошибок на нейтральных объектах и $m_{av} = 0$. Тогда, согласно Лемме 5.7,

$$H_{L_{av}}^{\ell_{av},0}(s_{av}(\varepsilon)) = \big[s_{av}(\varepsilon) \geqslant 0\big] = \big[\tfrac{\ell}{L}\big(n(a, \mathbb{X}) - \varepsilon k\big) - n(a, X_{av}) \geqslant 0\big] = \big[n(a, \mathbb{X}) \geqslant \varepsilon k\big].$$

Подставляя это выражение в (5.8), получаем (5.9). ■

Следствие 5.8.1. *Если в семействе A содержится алгоритм, корректный на всей генеральной выборке, то выражение* (5.9) *справедливо и для функционала $R_\varepsilon(\mu, \mathbb{X})$.*

§5.4 Функционал полного скользящего контроля

Рассмотрим функционал полного скользящего контроля (1.7). Принцип порождающих и запрещающих множеств также даёт для него точную оценку [11].

Теорема 5.9. *Если гипотеза 5.2 справедлива, то оценка полного скользящего контроля вычисляется по формуле*

$$C(\mu, \mathbb{X}) = \frac{1}{k} \sum_{a \in A} \sum_{v \in V_a} c_{av} \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell} \left(\frac{L_{av} - \ell_{av}}{L_{av}} n(a, \mathbb{X}) + \frac{\ell_{av}}{L_{av}} n(a, X'_{av}) \right). \tag{5.10}$$

Доказательство. Запишем определения E и ν, затем подставим (5.5) согласно гипотезе 5.2 и переставим знаки суммирования:

$$C(\mu,\mathbb{X}) = \frac{1}{C_L^\ell} \sum_{X\in[\mathbb{X}]^\ell} \sum_{a\in A} \big[\mu X{=}a\big] \frac{1}{k} \sum_{x_i\in\bar{X}} I(a,x_i) =$$

$$= \frac{1}{k} \sum_{a\in A} \sum_{v\in V_a} c_{av} \sum_{i=1}^{L} I(a,x_i) \underbrace{\mathsf{P}\big[X_{av}\subseteq X\big]\big[X'_{av}\subseteq\bar{X}\big]\big[x_i\in\bar{X}\big]}_{p(a,v,x_i)}.$$

Если $x_i \in X'_{av}$, то $\big[X'_{av}\subseteq\bar{X}\big]\big[x_i\in\bar{X}\big] = \big[X'_{av}\subseteq\bar{X}\big]$ и, согласно (5.7),

$$p(a,v,x_i) = \mathsf{P}\big[X_{av}\subseteq X\big]\big[X'_{av}\subseteq\bar{X}\big] = \frac{C_{L_{av}}^{\ell_{av}}}{C_L^\ell} = P_{av}.$$

Если $x_i \notin X'_{av}$, то $\big[X'_{av}\subseteq\bar{X}\big]\big[x_i\in\bar{X}\big] = \big[\{x_i\}\cup X'_{av}\subseteq\bar{X}\big]$ и, согласно (5.7),

$$p(a,v,x_i) = \mathsf{P}\big[X_{av}\subseteq X\big]\big[\{x_i\}\cup X'_{av}\subseteq\bar{X}\big] = \frac{C_{L_{av}-1}^{\ell_{av}}}{C_L^\ell} = P_{av}\frac{L_{av}-\ell_{av}}{L_{av}}.$$

Собирая вместе два случая, $x_i \in X'_{av}$ и $x_i \notin X'_{av}$, получим

$$p(a,v,x_i) = P_{av}\left(\big[x_i\in X'_{av}\big] + \big[x_i\notin X'_{av}\big]\frac{L_{av}-\ell_{av}}{L_{av}}\right).$$

Подставляя это выражение в сумму по i, получаем

$$\sum_{i=1}^{L} I(a,x_i)p(a,v,x_i) = P_{av}\left(n(a,X'_{av}) + \big(n(a,\mathbb{X}) - n(a,X'_{av})\big)\frac{L_{av}-\ell_{av}}{L_{av}}\right),$$

откуда вытекает требуемое равенство (5.10). Теорема доказана. ■

Резюме

Принцип порождающих и запрещающих множеств основан на гипотезе, что для каждого алгоритма можно указать множество *порождающих* объектов, которые обязаны быть в обучающей выборке, и множество *запрещающих* объектов, которых не должно быть в обучающей выборке, чтобы метод обучения выбрал именно данный алгоритм. Это довольно сильное предположение, и оно выполняется далеко не всегда. В общем случае для каждого алгоритма можно указать несколько пар порождающих и запрещающих множеств. Если они известны, то далее легко вычисляются вероятности получить каждый из алгоритмов, вероятность переобучения, и оценка полного скользящего контроля. Выбор системы порождающих и запрещающих множеств является искусством. Чем их меньше, и чем меньше их мощности, тем эффективнее будут вычисляться оценки.

В следующей лекции с помощью принципа порождающих и запрещающих множеств мы получим точные оценки вероятности переобучения для некоторых модельных семейств алгоритмов.

6 Монотонные цепи алгоритмов

Чтобы воспользоваться принципом порождающих и запрещающих множеств, необходимо конкретизировать метод обучения μ и семейство алгоритмов A. Мы начнём с метода минимизации эмпирического риска и монотонной цепи алгоритмов — самого простого модельного семейства, уже обладающего свойствами расслоения и связности. Чтобы подкрепить теоретическое исследование экспериментами, предлагается написать несложную программу для вычисления вероятности переобучения любых семейств, задаваемых непосредственно матрицей ошибок.

§6.1 Разновидности минимизации эмпирического риска

Будем полагать, что A — конечное множество, и все алгоритмы имеют попарно различные векторы ошибок. Обозначим через $A(X)$ множество алгоритмов с минимальным числом ошибок на обучающей выборке X:

$$A(X) = \operatorname{Arg}\min_{a\in A} n(a, X). \tag{6.1}$$

Определение 6.1. *Метод обучения μ называется минимизацией эмпирического риска, МЭР (empirical risk minimization, ERM), если $\mu X \in A(X)$ при всех $X \in [\mathbb{X}]^\ell$.*

Если множество $A(X)$ содержит более одного элемента, то возникает проблема неоднозначности выбора алгоритма. Рассмотрим сначала два крайних случая — когда выбирается наилучший или наихудший алгоритм из $A(X)$.

Определение 6.2. *Метод минимизации эмпирического риска μ называется оптимистичным, если $\mu X = \arg\min\limits_{a\in A(X)} n(a, \bar{X})$.*

Определение 6.3. *Метод минимизации эмпирического риска μ называется пессимистичным, если $\mu X = \arg\max\limits_{a\in A(X)} n(a, \bar{X})$.*

Оптимистичная и пессимистичная МЭР на практике не реализуемы, так как скрытую контрольную выборку $\bar{X}$ невозможно знать на этапе обучения. Теоретически они интересны тем, что дают нижнюю и верхнюю оценки вероятности переобучения. При любых других способах разрешения неоднозначности в методе МЭР вероятность переобучения гарантированно зажата между этими двумя оценками.

Более практичным представляется способ разрешения неоднозначности, основанный на случайном выборе алгоритма из множества $A(X)$.

Определение 6.4. *Метод минимизации эмпирического риска μ называется рандомизированным, если μX — это произвольный алгоритм, выбранный случайно и равновероятно из конечного множества алгоритмов $A(X)$.*

Рандомизация становится вторым независимым источником случайности в задаче статистического обучения. Если ранее предполагалось, что случайно только

разбиение $X \sqcup \bar{X}$, то теперь случаен также и выбор алгоритма a из множества $A(X)$. Соответствующим образом изменяется и определение вероятности переобучения:

$$Q_\varepsilon = \mathsf{E}\frac{1}{|A(X)|}\sum_{a\in A(X)}\big[\nu(a,\bar{X}) - \nu(a,X) \geqslant \varepsilon\big]. \tag{6.2}$$

Большинство получаемых далее оценок основаны либо на пессимистичной, либо на рандомизированной МЭР. Эксперименты показывают, что завышенность оценок пессимистичной МЭР невелика. Оптимистичные, пессимистичные и рандомизированные оценки сходятся друг к другу с ростом длины выборки L.

§6.2 Модельные семейства алгоритмов

Напомним, что нашим основным объектом исследования является бинарная матрица ошибок размера $L \times D$, порождаемая заданной выборкой $\mathbb{X} = \{x_1, \ldots, x_L\}$ и заданным семейством алгоритмов, из которого выбираются все алгоритмы с попарно различными векторами ошибок, $A = \{a_1, \ldots, a_D\}$.

Мы будем рассматривать *модельные семейства алгоритмов*, которые задаются непосредственно своими матрицами ошибок. Модельное семейство — это искусственный объект исследования, не связанный с какой-либо реальной выборкой. Матрицы ошибок модельных семейств обладают определёнными «регулярными структурами», что облегчает вывод точных комбинаторных оценок. Для некоторых модельных семейств удаётся строить примеры порождающих их выборок. Как правило, это весьма экзотические частные случаи. Реальные семейства в подавляющем большинстве случаев не обладают какой бы то ни было регулярной структурой.

Тем не менее, изучение модельных семейств представляет интерес по нескольким причинам. Во-первых, они позволяют исследовать влияние эффектов расслоения и связности на вероятность переобучения. Во-вторых, на них отрабатываются математические приёмы, которые могут оказаться полезными при получении оценок более общего вида. В-третьих, известны модельные семейства, обладающие теми же ключевыми свойствами, что и реальные — расслоением, связностью и размерностью. Вероятность переобучения таких семейств может довольно точно аппроксимировать вероятность переобучения реальных семейств, см., например, [5].

§6.3 Связные семейства алгоритмов

Определим расстояние между алгоритмами как *расстояние Хэмминга* между их векторами ошибок:

$$\rho(a,a') = \sum_{i=1}^{L}\big|I(a,x_i) - I(a',x_i)\big|, \quad \forall a,a' \in A.$$

Определение 6.5. *Конечное множество алгоритмов $A = \{a_1, \ldots, a_D\}$ называется цепью алгоритмов, если $\rho(a_{d-1}, a_d) = 1$ для всех $d = 2, \ldots, D$.*

Определение 6.6. *Множество алгоритмов A называется связным, если для любых $a, a' \in A$ в A существует цепь $a = a_1, a_2, \ldots, a_D = a'$.*

Определение 6.7. *Цепь алгоритмов $A = \{a_0, a_1, \ldots, a_D\}$ называется монотонной, если $n(a_d, \mathbb{X}) = m + d$ для всех $d = 0, \ldots, D$, при некотором $m \geqslant 0$. Алгоритм a_0 называется лучшим в цепи.*

В монотонной цепи $I(a_{d-1}, x_i) \leqslant I(a_d, x_i)$ для всех $x_i \in \mathbb{X}$, $d = 1, \ldots, D$.

В случае монотонной цепи объекты выборки можно перенумеровать так, чтобы

$$I(a_d, x_i) = \big[i \leqslant m + d\big], \quad i = 1, \ldots, L, \quad d = 0, \ldots, D.$$

Цепь алгоритмов может порождаться, в частности, однопараметрическим семейством классификаторов с непрерывной по параметру дискриминантной функцией. Если при непрерывном смещении параметра в сторону от оптимального значения число ошибок на полной выборке монотонно не убывает, то образуется монотонная цепь. Монотонная цепь — это одно из простейших модельных семейств, обладающее свойствами *расслоения* и *связности*.

Пример 6.1. Пусть A — семейство *линейных алгоритмов классификации* — параметрических отображений из $\mathbb{X} = \mathbb{R}^n$ в $\mathbb{Y} = \{-1, +1\}$ вида

$$a(x, w) = \operatorname{sign}(x_1 w_1 + \cdots + x_n w_n), \quad x = (x_1, \ldots, x_n) \in \mathbb{R}^n,$$

где параметр $w \in \mathbb{R}^n$ — направляющий вектор гиперплоскости, разделяющей пространство $\mathbb{R}^n$ на два полупространства — классы -1 и $+1$. Пусть функция потерь имеет вид $I(a, x) = \big[a(x, w) \neq y(x)\big]$, где $y(x)$ — истинная классификация объекта x, и множество объектов $\mathbb{X}$ линейно разделимо, т. е. существует вектор $w^* \in \mathbb{R}^n$, при котором алгоритм $a(x, w^*)$ не допускает ошибок на $\mathbb{X}$. Тогда множество алгоритмов

$$A_\delta = \big\{a(x, w^* + t\delta) \colon t \in [0, +\infty)\big\}$$

порождает монотонную цепь при любом $\delta \in \mathbb{R}^n$, за исключением, быть может, некоторого конечного множества векторов. При этом $m = 0$ в силу линейной разделимости.

§6.4 Эксперимент с цепями алгоритмов

Речь пойдёт о простом и очень наглядном эксперименте, который показывает, что эффекты расслоения и связности определяющим образом влияют на вероятность переобучения [60, 61]. Именно с этого эксперимента и началось активное развитие комбинаторной теории переобучения.

Эмпирическое оценивание вероятности переобучения по матрице ошибок. Алгоритм 6.1 вычисляет две эмпирические оценки вероятности переобучения: верхняя $\bar{Q}_\varepsilon$ соответствует пессимистичной МЭР, нижняя $\underline{Q}_\varepsilon$ — оптимистичной. Оценки вычисляются *методом Монте-Карло*, то есть путём усреднения по случайному подмножеству разбиений генеральной выборки $\mathbb{X} = X \sqcup \bar{X}$.

Алгоритм 6.1. Вычисление эмпирических оценок вероятности переобучения.

Вход: матрица ошибок $A = \{a_1, \ldots, a_D\}$; число разбиений N; порог ε;
Выход: верхняя оценка $\bar{Q}_\varepsilon$, нижняя оценка $\underline{Q}_\varepsilon$.

$\bar{Q}_\varepsilon := 0$; $\underline{Q}_\varepsilon := 0$;
для всех N случайных разбиений $X \sqcup \bar{X} = \mathbb{X}$
  $A(X) := \operatorname{Arg}\min_{a \in A} \nu(a, X)$;
  $\bar{a} := \arg\max_{a \in A(X)} \nu(a, \bar{X})$; $\bar{Q}_\varepsilon := \bar{Q}_\varepsilon + \frac{1}{N}\big[\delta(\bar{a}, X) \geqslant \varepsilon\big]$;
  $\underline{a} := \arg\min_{a \in A(X)} \nu(a, \bar{X})$; $\underline{Q}_\varepsilon := \underline{Q}_\varepsilon + \frac{1}{N}\big[\delta(\underline{a}, X) \geqslant \varepsilon\big]$;
вернуть $\bar{Q}_\varepsilon$, $\underline{Q}_\varepsilon$.

Алгоритм 6.2. Эффективное построение графиков зависимости вероятности переобучения от числа d первых алгоритмов в семействе $\{a_1, \ldots, a_D\}$.

Вход: матрица ошибок $A = \{a_1, \ldots, a_D\}$; число разбиений N; порог ε;
Выход: верхняя оценка $\bar{Q}_\varepsilon(d)$ и нижняя оценка $\underline{Q}_\varepsilon(d)$ для подмножества $\{a_1, \ldots, a_d\}$.

сгенерировать N случайных разбиений:
$X_n \sqcup \bar{X}_n = \mathbb{X}$, $n = 1, \ldots, N$;
алгоритм, выбранный при n-м разбиении пессимистичной и оптимистичной МЭР:
$\bar{a}_n$, $\underline{a}_n := a_1$, $n = 1, \ldots, N$;
$\bar{Q}_\varepsilon(1)$, $\underline{Q}_\varepsilon(1) := \frac{1}{N} \sum_{n=1}^{N} \big[\delta(a_1, X_n) \geqslant \varepsilon\big]$;
для всех $d := 2, \ldots, D$
  для всех $n := 1, \ldots, N$
    если $n(a_d, X_n) < n(\bar{a}_n, X_n)$ или
      $n(a_d, X_n) = n(\bar{a}_n, X_n)$ и $n(a_d, \bar{X}_n) > n(\bar{a}_n, \bar{X}_n)$ **то**
      $\bar{Q}_\varepsilon(d) := \bar{Q}_\varepsilon(d-1) - \frac{1}{N}\big[\delta(\bar{a}_n, X_n) \geqslant \varepsilon\big] + \frac{1}{N}\big[\delta(a_d, X_n) \geqslant \varepsilon\big]$;
      $\bar{a}_n := a_d$;
    если $n(a_d, X_n) < n(\underline{a}_n, X_n)$ или
      $n(a_d, X_n) = n(\underline{a}_n, X_n)$ и $n(a_d, \bar{X}_n) < n(\underline{a}_n, \bar{X}_n)$ **то**
      $\underline{Q}_\varepsilon(d) := \underline{Q}_\varepsilon(d-1) - \frac{1}{N}\big[\delta(\underline{a}_n, X_n) \geqslant \varepsilon\big] + \frac{1}{N}\big[\delta(a_d, X_n) \geqslant \varepsilon\big]$;
      $\underline{a}_n := a_d$;

Алгоритм 6.1 состоит из двух вложенных циклов. Внешний цикл перебирает N разбиений, внутренний цикл выбирает из A два алгоритма, минимизирующих эмпирический риск при данном разбиении — пессимистичный $\bar{a}$ и оптимистичный $\underline{a}$.

Поменяв местами два вложенных цикла перебора в Алгоритме 6.1, получим Алгоритм 6.2. Он позволяет за один запуск строить графики зависимостей вероятности переобучения от числа первых d алгоритмов в подсемействе $\{a_1, \ldots, a_d\}$. В этом случае становится важен порядок алгоритмов $a_1, \ldots, a_D$. Обычно будет предполагаться, что алгоритмы упорядочены по неубыванию числа ошибок на генеральной выборке,

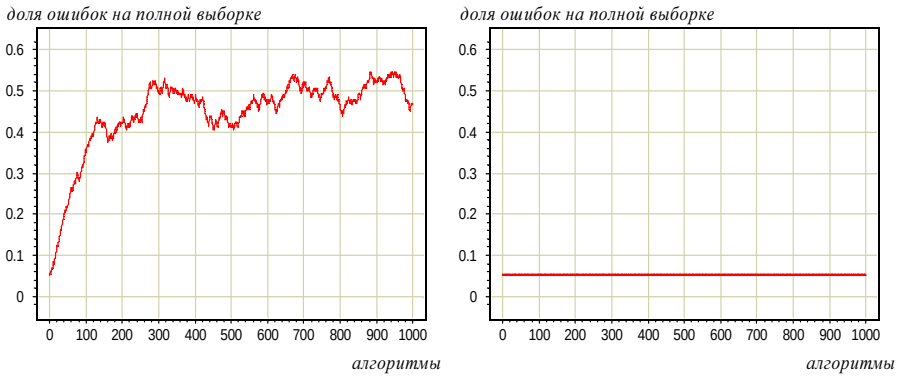


Рис. 6.1. Распределение алгоритмов по частоте ошибок на генеральной выборке в модельных цепях с расслоением и без расслоения. Зависимость $\nu(a_d, \mathbb{X})$ от d при $\ell = k = 100,\ m = 10$.

$n(a_1, \mathbb{X}) \leqslant \ldots \leqslant n(a_D, \mathbb{X})$. Число разбиений N рекомендуется брать не менее 1000, чтобы получать достаточно гладкие графики.

Эксперимент с четырьмя модельными семействами. Сгенерируем матрицы ошибок двух модельных цепей.

1. *Цепь с расслоением*. Лучший алгоритм a_1 допускает m ошибок на полной выборке. Каждый следующий вектор ошибок a_d получается из a_{d-1} путём инверсии одной случайно выбранной координаты. Если цепь достаточно длинная ($D \gg L$), то большинство алгоритмов допускают число ошибок m, близкое к $L/2$.

2. *Цепь без расслоения*. Лучший алгоритм a_1 также допускает m ошибок на полной выборке. Каждый вектор ошибок a_d также получается из a_{d-1} путём инверсии одной случайно выбранной координаты, но при нечётных d производятся только инверсии $0 \to 1$, а при чётных d — только $1 \to 0$. В результате число ошибок алгоритмов на полной выборке, чередуясь, принимает значения m и $m+1$.

На рис. 6.1 показаны зависимости частоты ошибок $\nu(a_d, \mathbb{X})$ от порядкового номера d алгоритма для цепи с расслоением (слева) и без расслоения (справа).

Для произвольной цепи $a_1, \ldots, a_D$ можно построить соответствующую ей *не-цепь* $a'_1, \ldots, a'_D$ с таким же распределением частот ошибок: $\nu(a'_d, \mathbb{X}) = \nu(a_d, \mathbb{X})$ для всех $d = 1, \ldots, D$. Чтобы соседние алгоритмы a'_{d-1}, a'_d существенно различались, компоненты векторов ошибок $\vec{a}'_d$ будем генерировать случайно и независимо, так, чтобы число единиц в векторе $\vec{a}'_d$ равнялось $n(a_d, \mathbb{X})$.

Итого, строятся четыре модельных семейства: цепь с расслоением, цепь без расслоения и две соответствующие им не-цепи. Все они задаются матрицами ошибок размера $L \times D$ и имеют одинаковые значения параметра m. Сопоставление этих четырёх случаев позволяет разделить влияние *связности* и *расслоения* на вероятность переобучения и ответить на вопрос — какое из этих двух свойств важнее.

На рис. 6.2 и рис. 6.3 показаны зависимости вероятности переобучения Q_ε от числа алгоритмов D для четырёх семейств, при $\ell = k = 100,\ \varepsilon = 0.05,\ m = 10$ и $m = 50$. Число разбиений в методе Монте-Карло $N = 10^4$. Условные обозначения на графиках: +Ц — цепь, −Ц — не-цепь, +Р — с расслоением, −Р — без расслоения.

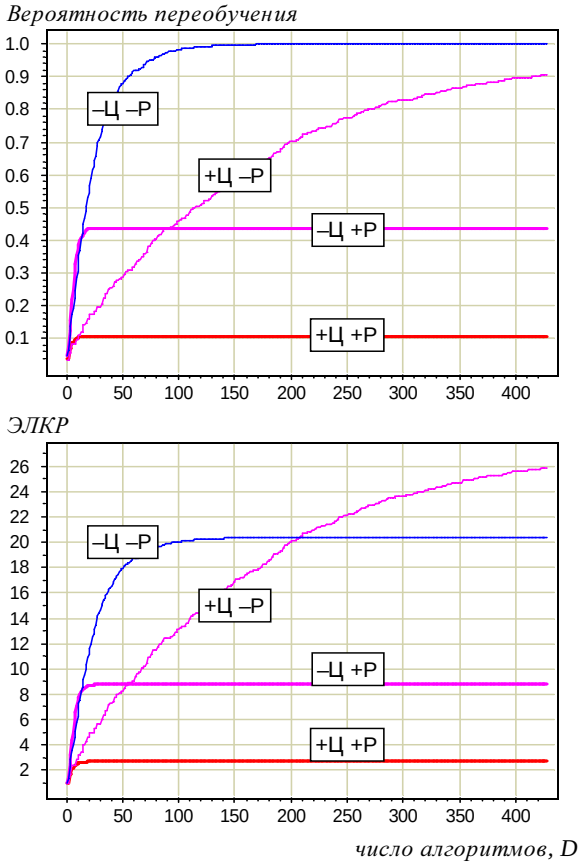


Рис. 6.2. Зависимость вероятности переобучения Q_ε и ЭЛКР $\hat{\Delta}$ от числа алгоритмов D. Простая задача: $\nu(a_1, \mathbb{X}) = 0.05$.

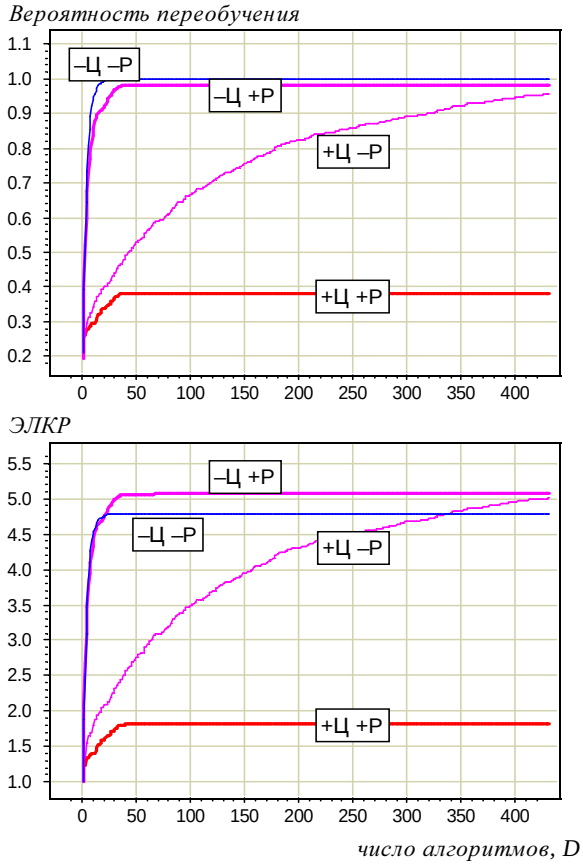


Рис. 6.3. Зависимость вероятности переобучения Q_ε и ЭЛКР $\hat{\Delta}$ от числа алгоритмов D. Трудная задача: $\nu(a_1, \mathbb{X}) = 0.25$.

Показаны также графики зависимости ЭЛКР — эффективного локального коэффициента разнообразия $\hat{\Delta}$ от числа алгоритмов D. ЭЛКР в данном случае определяется как отношение вероятности переобучения подсемейства $\{a_1, \ldots, a_D\}$ к вероятности переобучения одноэлементного подсемейства $\{a_1\}$.

Основные выводы.

1. Для семейств без расслоения и связности переобучение может оказаться значительным уже при нескольких десятках алгоритмов в семействе. При больших D только одновременное наличие расслоения и связности позволяет избежать сильного переобучения (нижние кривые на графиках). Таким образом, важны оба свойства.

2. Связность снижает темп роста зависимости $Q_\varepsilon(D)$.

3. Расслоение понижает уровень горизонтальной асимптоты $Q_\varepsilon(D)$, причём в большей степени для «лёгких задач» с меньшим значением m, рис. 6.2. В случае расслоения вероятность переобучения Q_ε может проходить существенно ниже единицы. Это означает, что вероятность выбрать в результате обучения алгоритм из верхних слоёв очень быстро падает с ростом номера слоя.

4. Для относительно простых задач, когда существует алгоритм с малым числом ошибок, расслоение существенно уменьшает вероятность переобучения, рис. 6.2. При увеличении сложности задачи влияние расслоения уменьшается, рис. 6.3.

5. По мере увеличения D как вероятность переобучения, так и ЭЛКР выходят на горизонтальную асимптоту и перестают зависеть от D. В то же время, VC-оценка линейна по D и вообще не имеет горизонтальной асимптоты — на графиках ЭЛКР VC-оценке соответствует прямая $\Delta(D) = D$. VC-оценка достигается только для нецепей и только при малых D; в данном эксперименте — при $D < 10$; при D порядка 20 она уже превосходит единицу.

§6.5 Монотонная цепь алгоритмов

Теорема 6.1. *Пусть $A = \{a_0, a_1, \ldots, a_D\}$ — монотонная цепь алгоритмов, $L \geqslant m + D$, $m = n(a_0, \mathbb{X})$, метод обучения μ является пессимистичной минимизацией эмпирического риска. Тогда в случае $D \geqslant k$*

$$Q_\varepsilon = \sum_{d=0}^{k} P_d H_{L-d-1}^{\ell-1,\,m}\left(s_d(\varepsilon)\right); \qquad P_d = \frac{C_{L-d-1}^{\ell-1}}{C_L^\ell},\ d = 0, \ldots, k;$$

в случае $D < k$

$$Q_\varepsilon = \sum_{d=0}^{D-1} P_d H_{L-d-1}^{\ell-1,\,m}\left(s_d(\varepsilon)\right) \ + \ P_D H_{L-D}^{\ell,\,m}\left(s_D(\varepsilon)\right);$$
$$P_d = \frac{C_{L-d-1}^{\ell-1}}{C_L^\ell},\ d = 0, \ldots, D-1; \qquad P_D = \frac{C_{L-D}^{\ell}}{C_L^\ell},$$

где $P_d = \mathsf{P}[\mu X = a_d]$ — вероятность получить алгоритм a_d; $s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$.

Доказательство. Перенумеруем объекты таким образом, чтобы каждый из алгоритмов $a_d,\ d = 1, \ldots, D$ допускал ошибку на объектах $x_1, \ldots, x_d$. Очевидно, лучший алгоритм a_0 не допускает ошибку ни на одном из этих объектов. Нумерация остальных объектов не имеет значения, так как алгоритмы не различимы на них.

Для наглядности представим выборку $\mathbb{X}$ разбитой на три блока:

$$\begin{array}{rccccccc}
 & x_1 & x_2 & x_3 & & x_D & & \overbrace{\qquad}^{m} \\
\vec{a}_0 = (& 0, & 0, & 0, & \ldots & 0, & 0, \ldots, 0, & 1, \ldots, 1 \ ); \\
\vec{a}_1 = (& 1, & 0, & 0, & \ldots & 0, & 0, \ldots, 0, & 1, \ldots, 1 \ ); \\
\vec{a}_2 = (& 1, & 1, & 0, & \ldots & 0, & 0, \ldots, 0, & 1, \ldots, 1 \ ); \\
\vec{a}_3 = (& 1, & 1, & 1, & \ldots & 0, & 0, \ldots, 0, & 1, \ldots, 1 \ ); \\
\ldots & & & & \ldots & & \ldots & \ldots \\
\vec{a}_D = (& 1, & 1, & 1, & \ldots & 1, & 0, \ldots, 0, & 1, \ldots, 1 \ );
\end{array}$$

При рассмотрении алгоритма a_d возможны три случая.

1. Если $k < d$, то число ошибок алгоритма a_d на объектах $\{x_1, \ldots, x_d\}$ превышает длину контрольной выборки. Часть ошибок обязательно окажется в обучающей

подвыборке X, и метод μ выберет другой алгоритм. В этом случае

$$[\mu X{=}a_d] = 0.$$

2. Если $d = D < k$, то метод μ выберет наихудший алгоритм в цепи a_D тогда и только тогда, когда все объекты $\{x_1, \ldots, x_D\}$ будут находиться в контрольной подвыборке $\bar{X}$. В этом случае

$$[\mu X{=}a_d] = [x_1, \ldots, x_D \in \bar{X}].$$

3. Во всех остальных случаях метод μ выберет алгоритм a_d, если только все объекты $\{x_1, \ldots, x_d\}$ будут находиться в контрольной подвыборке $\bar{X}$, а объект x_{d+1} — в обучающей подвыборке X. В этом случае

$$[\mu X{=}a_d] = [x_{d+1} \in X][x_1, \ldots, x_d \in \bar{X}].$$

Теперь можно применить Теорему 5.3.

Если $D \geqslant k$, то алгоритму a_d соответствуют следующие значения параметров (для упрощения обозначений вместо двойных индексов L_{a_d} будем использовать одинарные L_d): $L_d = L - d - 1$, $\ell_d = \ell - 1$, $m_d = m + d - d = m$, $s_d(\varepsilon) = \frac{\ell}{L}(m + d - \varepsilon k)$. Отсюда получаем утверждение теоремы для случая $D \geqslant k$.

Если $D < k$, то алгоритмам $a_0, \ldots, a_{D-1}$ соответствуют те же значения параметров, что и при $D \geqslant k$. Для наихудшего алгоритма a_D отличается только параметр $\ell_D = \ell$. Отсюда получаем утверждение теоремы для случая $D < k$. ■

Замечание 6.1. В ходе доказательства полезно проверить, что вероятности P_d вычислены корректно и в сумме дают единицу. Для случая $D \geqslant k$ проверка сводится к применению известного комбинаторного тождества:

$$\sum_{d=0}^{D} P_d = \sum_{d=0}^{k} P_d + \sum_{d=k+1}^{D} 0 = \frac{1}{C_L^\ell}\left(C_{L-1}^{\ell-1} + C_{L-2}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}\right) = 1.$$

Для случая $D < k$ то же самое тождество приходится применить дважды, заметив, что $C_{L-D}^{\ell} = C_{L-D-1}^{\ell-1} + \cdots + C_{\ell-1}^{\ell-1}$:

$$\sum_{d=0}^{D} P_d = \frac{1}{C_L^\ell}\left(C_{L-1}^{\ell-1} + \cdots + C_{L-D}^{\ell-1} + C_{L-D}^{\ell}\right) = 1.$$

Вычислительный эксперимент. Построим зависимость вероятности переобучения Q_ε от точности ε и длины цепи D. Заодно проверим полученную формулу, сравнив результат с эмпирической оценкой $\hat{Q}_\varepsilon$, вычисленной *методом Монте-Карло* по $N = 1000$ случайных разбиений. Графики на рис. 6.4 построены при $\ell = k = 100$ и $m = 20$, то есть когда лучший алгоритм допускает 10% ошибок на полной выборке.

Пессимистичная и рандомизированная МЭР дают почти одинаковые оценки Q_ε, оптимистичная — заметно заниженную оценку, см. левый график на рис. 6.4.

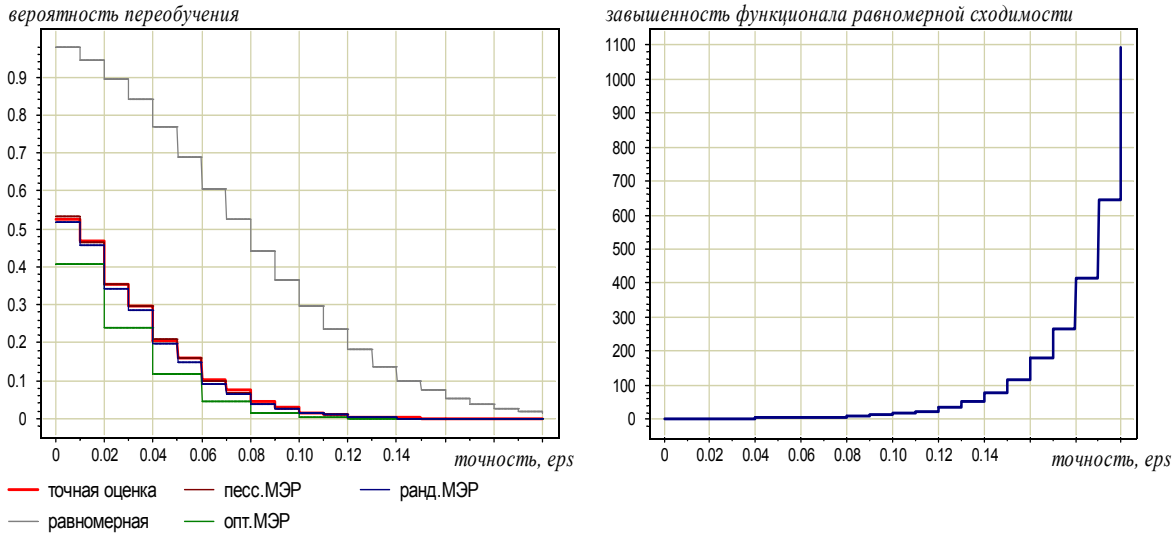


Рис. 6.4. Слева: зависимость оценок вероятности переобучения Q_ε от ε: точная оценка из Теоремы 6.1 и четыре оценки, вычисленные методом Монте-Карло по 1000 случайных разбиений: для пессимистичной, оптимистичной и рандомизированной МЭР. Верхняя кривая соответствует оценке по функционалу равномерной сходимости $\hat{P}_\varepsilon$. Справа: степень завышенности функционала равномерной сходимости $\hat{P}_\varepsilon/Q_\varepsilon$. Все графики построены при $\ell = k = 100,\ m = 20$.

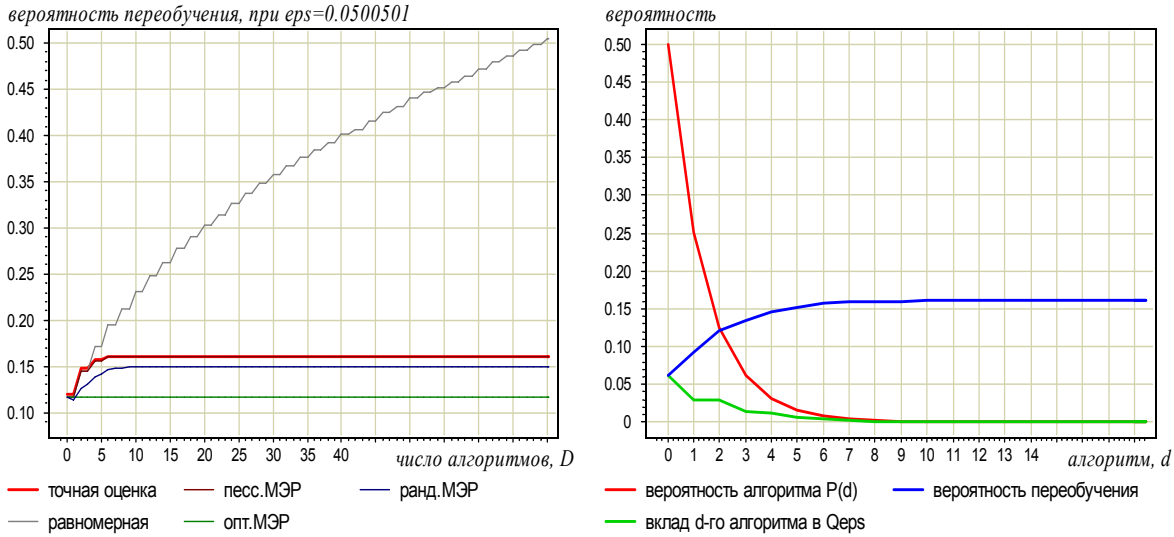


Рис. 6.5. Слева: зависимость вероятности переобучения Q_ε от числа алгоритмов D. Справа: вероятность получения каждого из алгоритмов $P_d = \mathsf{P}[\mu X{=}a_d]$, вклад каждого алгоритма в вероятность переобучения Q_ε, значение Q_ε для пессимистичной МЭР по подмножеству алгоритмов $\{a_0, \ldots, a_d\}$ как функция от числа алгоритмов d. Все графики построены при $\ell = k = 100,\ m = 20,\ \varepsilon = 0.05$.

В эксперименте вычислялась также эмпирическая оценка функционала равномерной сходимости,

$$\hat{P}_\varepsilon = \hat{\mathsf{P}}\Big[\max_{a\in A} \delta(a, X) \geqslant \varepsilon\Big].$$

Он является завышенной верхней оценкой вероятности переобучения, $Q_\varepsilon \leqslant P_\varepsilon$, см. стр. 27. Правый график на рис. 6.4 показывает, что эта оценка может быть завышенной в сотни раз.

Левый график на рис. 6.5 показывает, что с ростом числа алгоритмов в монотонной цепи функционал равномерной сходимости P_ε продолжает возрастать, тогда как вероятность переобучения Q_ε после 5–8 алгоритмов выходит на горизонтальную асимптоту. Согласно Теореме 3.1, оценка равномерной сходимости завышена из-за того, что она не учитывает эффект расслоения. Поэтому кривую $\hat{P}_\varepsilon$ (верхняя кривая на левом графике рис. 6.5) можно рассматривать как оценку вероятности переобучения для *цепи без расслоения*. Только совместное проявление эффектов расслоения и связности понижает вероятность переобучения до приемлемо малых значений. Этот же вывод был сделан выше в экспериментах со случайными цепями.

Вкладом $Q_\varepsilon(a)$ алгоритма a в вероятность переобучения Q_ε будем называть слагаемое под знаком суммы $\sum\limits_{a\in A}$ в общей формуле вероятности переобучения (5.8):

$$Q_\varepsilon = \sum_{a\in A} \underbrace{\sum_{v\in V_a} c_{av} P_{av} H_{L_{av}}^{\ell_{av},\, m_{av}}\left(s_{av}(\varepsilon)\right)}_{Q_\varepsilon(a)}.$$

Рис. 6.5 (справа) показывает, что существенные вклады в вероятность переобучения вносят только алгоритмы нескольких нижних слоёв. Это справедливо не только для монотонных цепей, но и для многих других семейств алгоритмов.

Резюме

Модельные семейства алгоритмов задаются непосредственно матрицами ошибок, а не реальными выборками. Монотонная цепь алгоритмов — это простое, и в то же время нетривиальное, модельное семейство, обладающее свойствами *расслоения* и *связности*.

Метод порождающих и запрещающих множеств даёт точную оценку вероятности переобучения для монотонной цепи алгоритмов

Монотонная цепь алгоритмов почти не переобучается. Этот факт служит косвенным обоснованием для процедур одномерной оптимизации, которые часто применяются в машинном обучении для выбора одного критически важного параметра по отложенной выборке (hold-out model selection), например, константы регуляризации, ширины окна сглаживания, и т. п.

В следующей лекции мы введём понятие графа расслоения–связности семейства алгоритмов и получим верхнюю оценку вероятности переобучения, применимую к произвольным семействам, как модельным, так и реальным. В общем случае это именно верхняя оценка, однако для монотонных цепей и некоторых других модельных семейств алгоритмов она является точной.

Упражнения

В следующих упражнениях предлагается вывести точную оценку вероятности переобучения Q_ε для некоторого модельного семейства алгоритмов A, в котором все векторы ошибок попарно различны. Предполагается что метод обучения μ является пессимистичной минимизацией эмпирического риска.

Задача 6.1 (1). Множество A есть m-й слой L-мерного булева куба. Состоит из всех C_L^m алгоритмов, допускающих ровно m ошибок на полной выборке $\mathbb{X}$.

Задача 6.2 (2). Интервал ранга m в L-мерном булевом кубе. Задаётся разбиением всех объектов на три группы: m_0 «внутренних» объектов, на которых ни один из алгоритмов не допускает ошибок; m_1 «шумовых» объектов, на которых все алгоритмы допускают ошибки; и m «пограничных» объектов, на которых реализуются все 2^m вариантов допустить ошибки. Других объектов нет: $m_0 + m_1 + m = L$.

Задача 6.3 (1). Подмножество интервала ранга m в L-мерном булевом кубе, состоящее из тех и только тех алгоритмов, которые допускают ровно t ошибок на пограничных объектах (t-й слой интервала состоит из C_m^t алгоритмов).

Задача 6.4 (1). Подмножество интервала ранга m в L-мерном булевом кубе, состоящее из тех и только тех алгоритмов, которые допускают не более t ошибок на пограничных объектах (t нижних слоёв интервала, $C_m^0 + \cdots + C_m^t$ алгоритмов).

Задача 6.5 (1). $A = \{a_0, a_1, \ldots, a_D\}$ — единичная окрестность лучшего алгоритма a_0. Содержит D алгоритмов с попарно различными векторами ошибок, каждый из которых допускает на одну ошибку больше, чем a_0.

Задача 6.6 (3). $A = \{a_0, a_1, \ldots, a_D, a'_1, \ldots, a'_D\}$ — симметричная унимодальная цепь алгоритмов. *Левая ветвь* $a_0, a_1, \ldots, a_D$ и *правая ветвь* $a_0, a'_1, \ldots, a'_D$ являются монотонными цепями с общим алгоритмом a_0, который называется *лучшим в цепи*.

Задача 6.7 (4). $A = \{a_0, a_1, \ldots, a_D, a'_1, \ldots, a'_{D'}\}$ — несимметричная унимодальная цепь алгоритмов, $D \neq D'$. *Левая ветвь* $a_0, a_1, \ldots, a_D$ и *правая ветвь* $a_0, a'_1, \ldots, a'_{D'}$ являются монотонными цепями с общим *лучшим* алгоритмом a_0.

Задача 6.8 (5*). Выпрямленная цепь с уровнем ошибок m — множество векторов ошибок $a_0, a_1, \ldots, a_D$ такое, что $n(a_d, \mathbb{X}) = m + (d \bmod 2)$ и хэммингово расстояние $\rho(a_{d'}, a_d) = |d' - d|$ для любых $d, d' = 0, \ldots, D$.

Задача 6.9 (5*). Матрица ошибок m-диагональна, то есть алгоритм a_d допускает ошибки на объектах $x_{d+1}, \ldots, x_{d+m}$, $d = 0, \ldots, D$.

Задача 6.10 (10*). Произвольная цепь алгоритмов.

Практикум

В практическом задании предлагается реализовать Алгоритм 6.2 и использовать его для проверки теоретических оценок и исследования новых семейств.

Задача 6.11 (5). Написать программу, позволяющую:

- генерировать модельные семейства алгоритмов $A = \{a_1, \ldots, a_D\}$ в виде бинарной матрицы ошибок размера $L \times D$ (в матрице ошибок не должно быть одинаковых векторов ошибок); легко заменять генераторы данных;
- вычислять точные верхние и нижние оценки вероятности переобучения, если соответствующие формулы известны;
- вычислять эмпирические оценки вероятности переобучения методом Монте-Карло, т. е. по случайному подмножеству из N разбиений $(X, \bar{X})$; требуется вычислять три оценки, соответствующие трём стратегиям выбора алгоритма в случаях неоднозначного минимума эмпирического риска:
  — верхняя оценка $\bar{Q}_\varepsilon$ для пессимистичного $\mu_{\text{пес}}$ (худший из лучших);
  — нижняя оценка $\underline{Q}_\varepsilon$ для оптимистичного $\mu_{\text{опт}}$ (лучший из лучших);
  — средняя оценка $\hat{Q}_\varepsilon$ для рандомизированного $\mu_{\text{ран}}$ (случайный из лучших);
- строить графики, откладывая по оси X число первых d алгоритмов (либо, как вариант, номер слоя m), по оси Y:
  — эмпирические оценки $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon$ для подмножества $A(d) = \{a_1, \ldots, a_d\}$;
  — точные значения $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon$ для подмножества $A(d)$ (если известны);
  если по оси X откладывается m, то по оси Y дополнительно откладывать:
  — число алгоритмов в m-м слое;
  — доля разбиений, на которых $n(\mu_{\text{пес}} X, \mathbb{X}) = m$;
  — доля разбиений, на которых $n(\mu_{\text{опт}} X, \mathbb{X}) = m$;
- строить графики, в которых по оси Y откладываются точные (если известны) и эмпирические значения $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon$, по оси X:
  — число ошибок лучшего алгоритма m;
  — длина обучения ℓ, при одновременном росте длины контроля $\ell = k$;
  — длина обучения ℓ при фиксированной длине контроля k;
  — длина контроля k при фиксированной длине обучения ℓ.

Следующая серия задач направлена на экспериментальную проверку оценок и исследование зависимостей $\bar{Q}_\varepsilon, \underline{Q}_\varepsilon, \hat{Q}_\varepsilon$ от параметров модельного семейства.

Задача 6.12 (3). Монотонная цепочка с параметрами m и D.

Задача 6.13 (3). Унимодальная цепочка с параметрами m и D.

Задача 6.14 (3). Единичная окрестность с параметрами m и D.

Задача 6.15 (8). Интервал булева куба с параметрами m_0, m_1, m.

Следующий эксперимент направлен на проверку гипотезы, что монотонная цепочка и цепочка случайных инверсий [61] ведут себя практически одинаково с точки зрения переобучения.

Задача 6.16 (3). Сравнить, представив на одном графике, точные значения $\bar{Q}_\varepsilon$ для монотонной цепочки, их эмпирические оценки и эмпирические оценки $\bar{Q}_\varepsilon$ для цепочки случайных инверсий при одинаковых m. Как меняются различия между $\bar{Q}_\varepsilon$ монотонной цепочки и цепочки случайных инверсий с ростом ℓ и m?

7 Оценки расслоения–связности

Первые оценки вероятности переобучения для произвольного множества алгоритмов A, учитывавшие расслоение и связность, были получены в 2009 году независимо Д. Кочедыковым и И. Решетняком. В [62] была предложена оценка, основанная на рекуррентном вычислении порождающих и запрещающих множеств при поочерёдном добавлении алгоритмов в A. Её доказательство было довольно громоздким; кроме того, требовалось, чтобы в A существовал корректный алгоритм, не допускающий ошибок на генеральной выборке. Позже оценка была улучшена, ограничение корректности снято, и найдено простое доказательство, которое приводится ниже.

§7.1 Граф расслоения–связности

Напомним основные обозначения:
$\mathbb{X} = \{x_1, \ldots, x_L\}$ — конечное множество объектов;
A — множество алгоритмов;
$I(a, x)$ — индикатор ошибки алгоритма $a \in A$ на объекте $x \in \mathbb{X}$;
$n(a, X)$ — число ошибок алгоритма a на выборке $X \subseteq \mathbb{X}$;
$\rho(a, b)$ — хэммингово расстояние между векторами ошибок алгоритмов a и b;
$A_m = \big\{a \in A \colon n(a, \mathbb{X}) = m\big\}$ — *m-й слой* множества A.

Будем полагать, что все векторы ошибок $\vec{a} = \big(I(a, x_i)\big)_{i=1}^{L}$, порождаемые алгоритмами a из A, попарно различны. Если это не так, то алгоритмы, соответствующие дублирующим векторам ошибок, исключим из множества A.

Введём на A естественное отношение порядка: $a \leqslant b$ тогда и только тогда, когда $I(a, x) \leqslant I(b, x)$ для всех $x \in \mathbb{X}$. Определим $a < b$ если $a \leqslant b$ и $a \neq b$.

Если $a < b$ и при этом $\rho(a, b) = 1$, то будем говорить, что a *предшествует* b и записывать $a \prec b$. Очевидно, что $n(a, \mathbb{X}) + 1 = n(b, \mathbb{X})$.

Определение 7.1. *Графом расслоения–связности множества алгоритмов A будем называть направленный граф $\langle A, E\rangle$ с множеством рёбер $E = \big\{(a, b) \colon a \prec b\big\}$.*

Граф расслоения–связности является многодольным, доли соответствуют слоям A_m, рёбрами могут соединяться только алгоритмы соседних слоёв.

Каждому ребру $a \prec b$ графа расслоения–связности соответствует один и только один объект $x_{ab} \in \mathbb{X}$, такой, что $I(a, x_{ab}) = 0$ и $I(b, x_{ab}) = 1$.

Пример 7.1. На рис. 7.1 показан граф расслоения–связности, порождаемый семейством линейных алгоритмов классификации на выборке длины $L = 10$. Начальный фрагмент его матрицы ошибок приводился на рис. 1.2, стр. 8. Выборка линейно разделима, поэтому в графе имеется нулевой слой, состоящий из единственной вершины, соответствующей нулевому вектору ошибок. Первый слой образуется 5 алгоритмами с одной ошибкой, второй слой — 8 алгоритмами с двумя ошибками, и т. д.

В типичном случае граф расслоения–связности изоморфен графу транзитивной редукции отношения порядка $\leqslant$, называемому также *диаграммой Хáссе*. Отличие

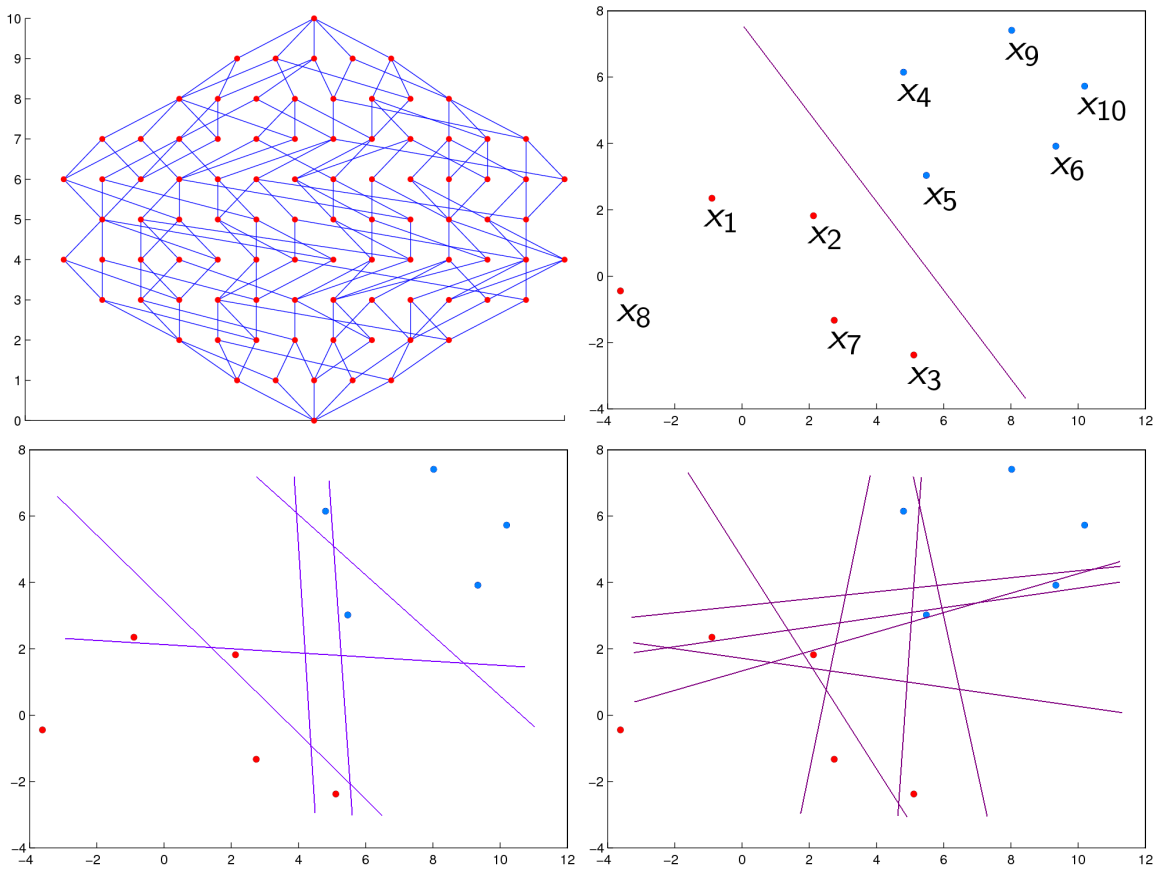


Рис. 7.1. Пример графа расслоения–связности (вверху слева; по вертикальной оси отложены номера слоёв), порождаемого семейством линейных алгоритмов классификации на выборке из 10 объектов, по 5 объектов каждого класса (вверху справа). Первый слой образуется 5 алгоритмами с одной ошибкой (внизу слева), второй слой — 8 алгоритмами с двумя ошибками (внизу справа), и т. д.

в том, что в графе $\langle A, E\rangle$ рёбрами соединяются только алгоритмы, отличающиеся на одном объекте, тогда как в диаграмме Хассе рёбрами соединяются также и алгоритмы a, b, отличающиеся более чем на одном объекте, если не существует такого $c \in A$, что $a < c < b$. В общем случае граф расслоения–связности является подграфом диаграммы Хассе естественного отношения порядка на множестве бинарных векторов ошибок всех алгоритмов семейства A.

§7.2 Оценки расслоения–связности

Ослабление метода порождающих и запрещающих множеств. Напомним, что для получения точных оценок обобщающей способности мы записывали необходимое и достаточное условие (5.5) того, что алгоритм $a \in A$ выдаётся методом μ в результате обучения по выборке $X \in [\mathbb{X}]^\ell$:

$$\big[\mu X{=}a\big] = \sum_{v \in V_a} c_{av} \big[X_{av} \subseteq X\big] \big[X'_{av} \subseteq \bar{X}\big].$$

Ослабим это условие.

Во-первых, ограничимся необходимым условием, что равносильно замене равенства оценкой сверху.

Во-вторых, чтобы упростить оценки, для каждого алгоритма будем искать только одну пару из порождающего и запрещающего множества, и полагать $c_{av} = 1$.

Гипотеза 7.1. *Пусть множество A, выборка $\mathbb{X}$ и метод μ таковы, что для каждого алгоритма $a \in A$ можно указать порождающее множество $X_a \subset \mathbb{X}$ и запрещающее множество $X'_a \subset \mathbb{X}$, удовлетворяющие условиям*

$$\big[\mu X{=}a\big] \leqslant \big[X_a \subseteq X\big]\big[X'_a \subseteq \bar{X}\big], \quad \forall X \in [\mathbb{X}]^\ell. \tag{7.1}$$

Если гипотеза 7.1 верна, то справедлива верхняя оценка, аналогичная точной оценке из Теоремы 5.3 (доказательство тривиально воспроизводится шаг за шагом):

$$\begin{aligned} Q_\varepsilon \leqslant \bar{Q}_\varepsilon &= \sum_{a\in A} \bar{P}_a H_{L_a}^{\ell_a,\, m_a}\left(s_a(\varepsilon)\right); \quad \bar{P}_a = \frac{C_{L_a}^{\ell_a}}{C_L^\ell}; \\ L_a &= L - |X_a| - |X'_a|; \\ \ell_a &= \ell - |X_a|; \\ m_a &= n(a, \mathbb{X}\backslash X_a \backslash X'_a); \\ s_a(\varepsilon) &= \tfrac{\ell}{L}\big(n(a,\mathbb{X}) - \varepsilon k\big) - n(a, X_a). \end{aligned} \tag{7.2}$$

Величина $\bar{P}_a$ является верхней оценкой вероятности $P_a = \mathsf{P}[\mu X = a]$ получить алгоритм a в результате обучения. Поскольку $\sum\limits_{a\in A} P_a = 1$ и $\bar{P}_a \geqslant P_a$, величина $\sum\limits_{a\in A} \bar{P}_a$, скорее всего, окажется большей единицы. Она будет приблизительно равна степени завышенности оценки вероятности переобучения $\bar{Q}_\varepsilon$.

Верхняя оценка вероятности переобучения. Для каждого алгоритма $a \in A$ определим два множества объектов: X_a — множество объектов x_{ab}, соответствующих всевозможным рёбрам графа $(a, b) \in E$, исходящим из a:

$$X_a = \big\{x_{ab} \in \mathbb{X}\colon a \prec b\big\}, \tag{7.3}$$

и X'_a — множество объектов $x \in \mathbb{X}$, таких, что a ошибается на x и существует лучший алгоритм $b \leqslant a$, который не ошибается на x:

$$X'_a = \big\{x_{bc} \in \mathbb{X}\colon b \prec c \leqslant a\big\}. \tag{7.4}$$

Лемма 7.1. *Если μ — метод пессимистичной минимизации эмпирического риска, то множества X_a (7.3) и X'_a (7.4) являются, соответственно, порождающим и запрещающим для алгоритма a в смысле Гипотезы 7.1.*

Доказательство. Докажем от противного, что если $\mu X = a$, то $X_a \subseteq X$ и $X'_a \subseteq \bar{X}$.

Допустим, что найдётся $x_{ab} \in X_a$, не лежащий в X. Тогда $n(a, X) = n(b, X)$, поскольку векторы ошибок a и b отличаются только на объекте x_{ab}. В то же время,

$n(a, \mathbb{X}) + 1 = n(b, \mathbb{X})$, поэтому для метода μ, в силу его пессимистичности, выбор алгоритма b по выборке X будет предпочтительнее, чем a, что противоречит условию $\mu X = a$. Значит, $X_a \subseteq X$.

Допустим теперь, что найдётся $x \in X'_a$, лежащий в X. Тогда существует $b \in A$, для которого $n(b, X) < n(a, X)$. Поскольку метод μ минимизирует эмпирический риск, выбор алгоритма b будет предпочтительнее, чем a, что противоречит условию $\mu X = a$. Значит, $X'_a \subseteq \bar{X}$.

Лемма доказана. ■

Определение 7.2. *Верхней связностью $q(a)$ (upper connectivity) алгоритма $a \in A$ будем называть число рёбер графа, исходящих из вершины a:*

$$q(a) = |X_a| = \#\big\{x_{ab} \in \mathbb{X} \colon a \prec b\big\}.$$

Определение 7.3. *Нижней связностью $d(a)$ (lower connectivity) алгоритма $a \in A$ будем называть число рёбер графа, входящих в вершину a:*

$$d(a) = |D_a| = \#\big\{x_{ba} \in \mathbb{X} \colon b \prec a\big\}.$$

Связность $q(a)$ (или $d(a)$) есть реализуемое семейством A число способов изменить алгоритм a так, чтобы он стал делать на одну ошибку больше (или меньше). Связность можно интерпретировать как число степеней свободы семейства A в локальной окрестности алгоритма $a \in A$.

Определение 7.4. *Неоптимальностью $r(a)$ (inferiority) алгоритма $a \in A$ будем называть число объектов $x \in \mathbb{X}$, на которых алгоритм a ошибается, при том, что существует алгоритм $b \in A$, лучший, чем a (то есть $b \leqslant a$), не ошибающийся на x:*

$$r(a) = |X'_a| = \#\big\{x_{bc} \in \mathbb{X} \colon b \prec c \leqslant a\big\}.$$

В терминах графа расслоения–связности $r(a)$ есть число различных объектов x_{bc}, соответствующих всевозможным рёбрам (b, c) на путях, ведущих к вершине a.

Справедливо неравенство $d(a) \leqslant r(a) \leqslant n(a, \mathbb{X})$.

Равенство $r(a) = d(a)$ достигается на всех алгоритмах двух самых нижних слоёв.

Равенство $r(a) = n(a, \mathbb{X})$ достигается в случае, когда существует корректный алгоритм $a_0 \in A$: $n(a_0, \mathbb{X}) = 0$.

Теорема 7.2 (оценка расслоения–связности). *Пусть μ — метод пессимистичной минимизации эмпирического риска, векторы ошибок всех алгоритмов из A попарно различны. Тогда имеет место верхняя оценка вероятности переобучения:*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C^{\ell-q}_{L-q-r}}{C^\ell_L} H^{\ell-q,\,m-r}_{L-q-r}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \tag{7.5}$$

где $q = q(a)$ — верхняя связность, $r = r(a)$ — неоптимальность, $m = n(a, \mathbb{X})$ — число ошибок алгоритма a на генеральной выборке.

Доказательство. Данная оценка следует непосредственно из общей оценки (7.2) и Леммы 7.1, если заметить, что произвольный алгоритм $a \in A$ ошибается на всех объектах запрещающего множества X'_a и не ошибается на всех объектах порождающего множества X_a: $|X_a| = q(a),\ n(a, X_a) = 0,\ |X'_a| = n(a, X'_a) = r(a)$. ■

Рассмотрим некоторые свойства оценки расслоения–связности.

1. Благодаря комбинаторному сомножителю $C^{\ell-q}_{L-q-r}/C^\ell_L$ вклад каждого алгоритма a в оценку Q_ε экспоненциально убывает с ростом неоптимальности r и связности q. Отсюда следуют два важных для практики вывода. Во-первых, связные семейства менее подвержены переобучению. Во-вторых, только нижние слои вносят существенный вклад в переобучение. Благодаря последнему обстоятельству становится возможным эффективное приближённое вычисление $\bar{Q}_\varepsilon$ по слоям снизу вверх.

2. Если пренебречь расслоением и связностью, положив $r = q = 0$ для каждого $a \in A$ в формуле (7.5), то получится лучшая из VC-оценок (3.4):

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} H_L^{\ell,\, m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right).$$

3. Оценка расслоения–связности является достижимой. Неравенство $Q_\varepsilon \leqslant \bar{Q}_\varepsilon$ переходит в равенство, когда условие (7.1) является равенством. Другими словами, когда условие $[\mu X = a]$, выраженное с помощью единственной пары множеств (X_a, X'_a) для любого алгоритма $a \in A$, является не только необходимым, но и достаточным. Это так, в частности, для монотонной цепи алгоритмов (стр. 56) и её многомерного обобщения — монотонной сети алгоритмов, которая будет рассмотрена ниже.

4. Оценка расслоения–связности принимает особенно простой вид, когда в семействе A существует корректный алгоритм a_0: $n(a_0, \mathbb{X}) = 0$. Такие семейства будем называть *корректными* относительно выборки $\mathbb{X}$.

Теорема 7.3. *Если μ — метод пессимистичной минимизации эмпирического риска и семейство A корректно, то*

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C^{\ell-q}_{L-q-m}}{C^\ell_L}\big[m \geqslant \varepsilon k\big], \tag{7.6}$$

где $q = q(a),\ m = n(a, \mathbb{X})$.

Доказательство. Если в A существует корректный алгоритм a_0, то неоптимальность любого алгоритма $a \in A$ равна числу его ошибок, $r(a) = n(a, \mathbb{X})$. Поэтому в (7.5) можно подставить $m - r = 0$. Тогда, согласно Лемме 5.7 (стр. 48), гипергеометрическое распределение вырождается: $H^{\ell-q,\,0}_{L-q-m}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right) = [m \geqslant \varepsilon k]$. Отсюда вытекает утверждение теоремы. ■

§7.3 Профиль расслоения–связности

В ряде случаев общая оценка (7.5) приводится к более удобному виду.

Определение 7.5. *Профилем расслоения–связности множества алгоритмов A называется матрица $\big(\Delta_{mq}\big)$ размера $(L{+}1){\times}(L{+}1)$, где Δ_{mq} — число алгоритмов в m-м слое со связностью q:*

$$\Delta_{mq} = \sum_{a\in A} \big[n(a,\mathbb{X}) = m\big]\,\big[q(a) = q\big].$$

Теорема 7.4. *Пусть граф расслоения–связности имеет исток — единственный алгоритм a_0, от которого можно добраться по рёбрам до любого другого алгоритма и $m_0 = n(a_0,\mathbb{X})$. Тогда справедлива верхняя оценка вероятности переобучения*

$$Q_\varepsilon \leqslant \sum_{m=m_0}^{L} \sum_{q=0}^{L} \Delta_{mq} \frac{C^{\ell-q}_{L-q-m+m_0}}{C^{\ell}_{L}} H^{\ell-q,\ m_0}_{L-q-m+m_0}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right). \tag{7.7}$$

Доказательство. Возьмём произвольный алгоритм $a \in A$. В силу единственности истока все запрещающие объекты из X'_a находятся на рёбрах графа, составляющих всевозможные пути между вершинами a_0 и a. Поскольку все эти пути имеют общее начало a_0 и общий конец a, число объектов в X'_a равно в точности длине пути $n(a,\mathbb{X}) - m_0$. С другой стороны, оно же равно $r(a)$. Значит,

$$Q_\varepsilon \leqslant \sum_{m=m_0}^{L} \sum_{a\in A_m} \frac{C^{\ell-q(a)}_{L-q(a)-m+m_0}}{C^{\ell}_{L}} H^{\ell-q(a),\,m_0}_{L-q(a)-m+m_0}\left(\tfrac{\ell}{L}(m-\varepsilon k)\right).$$

Перегруппировав слагаемые и воспользовавшись определением профиля расслоения–связности, получим утверждение теоремы. ■

Теорема 7.5. *Пусть верны предположения предыдущей теоремы и исток является корректным алгоритмом, $m_0 = n(a_0,\mathbb{X}) = 0$. Тогда справедлива верхняя оценка вероятности переобучения*

$$Q_\varepsilon \leqslant \sum_{m=\lceil\varepsilon k\rceil}^{L} \sum_{q=0}^{L} \Delta_{mq} \frac{C^{\ell-q}_{L-q-m}}{C^{\ell}_{L}}. \tag{7.8}$$

Доказательство. В оценку (7.7) подставим $m_0 = 0$. Гипергеометрическое распределение вырождается согласно Лемме 5.7, откуда и получаем оценку (7.8). ■

Существование истока — довольно сильное предположение, и в большинстве практических ситуаций оно не выполняется. Предположение корректности ещё сильнее, поскольку вводится дополнительное требование, чтобы исток был корректным алгоритмом, $m_0 = n(a_0,\mathbb{X}) = 0$.

Гипотеза о сепарабельности профиля расслоения–связности. На рис. 7.2 показаны графики зависимости[9] Δ_{mq} от m и q, для множества линейных алгоритмов классификации и линейно разделимых двумерных выборок длины $L = 20,\ 50,\ 100$.

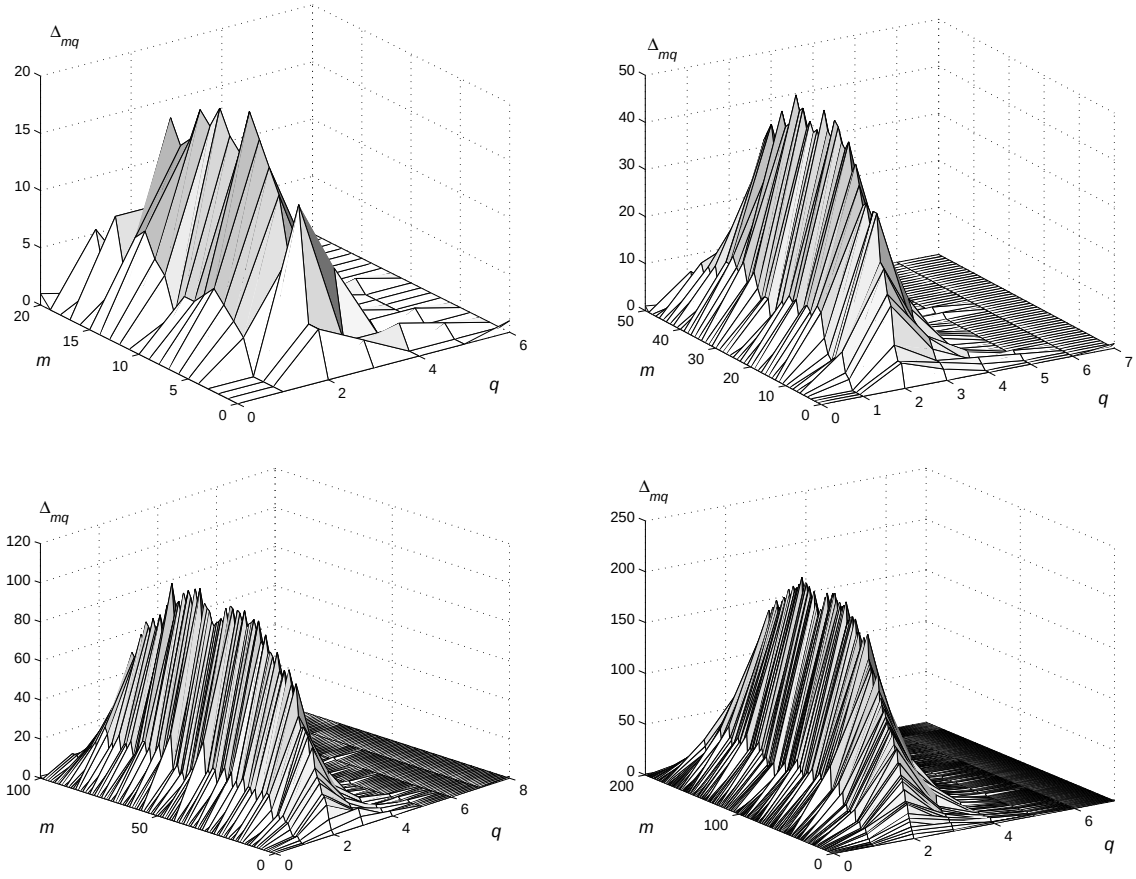


Рис. 7.2. Профили расслоения–связности для двумерных выборок длины $L = 20$, 50, 100, 200. Профиль Δ_{mq} — это количество алгоритмов с числом ошибок m на генеральной выборке и связностью q.

Видно, что профиль связности концентрируется в точке $q = 2$, что совпадает с размерностью пространства. С увеличением длины выборки доминирование данной компоненты профиля только усиливается.

Глядя на графики, можно выдвинуть гипотезу, что профиль расслоения–связности Δ_{mq} является с высокой точностью *сепарабельным*:

$$\Delta_{mq} \lessapprox \Delta_m \lambda_q,$$

где Δ_m — коэффициент разнообразия m-го слоя, λ_q — доля алгоритмов, имеющих связность q. Очевидно, выполняется условие нормировки $\lambda_0 + \cdots + \lambda_L = 1$.

Вектор $(\Delta_m)_{m=0}^{L}$ предлагается называть *профилем расслоения*, а вектор $(\lambda_q)_{q=0}^{L}$ — *профилем связности* множества алгоритмов A.

Доказательство гипотезы сепарабельности пока остаётся открытой проблемой.

---

[9]Вычислительные эксперименты выполнены Ильёй Решетняком.

Резюме

Метод порождающих и запрещающих множеств существенно упрощается, если для каждого алгоритма записать лишь необходимые условия того, что он будет получен в результате обучения. При этом вместо точных оценок вероятности переобучения получаются верхние оценки. Они существенно точнее VC-оценок, поскольку учитывают структуру графа расслоения–связности. Вклад алгоритма в вероятность переобучения убывает экспоненциально с ростом связности и номера слоя, в котором находится алгоритм. Отсюда следует, что связные семейства менее склонны к переобучению, а для приближённого вычисления вероятности переобучения, возможно, будет достаточно взять алгоритмы из нескольких нижних слоёв.

В следующей лекции мы применим оценку расслоения–связности к семейству конъюнктивных логических правил над вещественными признаками, которые широко используются в логических алгоритмах классификации.

Упражнения

Задача 7.1 (3*). Какие ограничения необходимо наложить на семейство A, чтобы условие (7.1) было не только необходимым, но и достаточным, следовательно, оценка расслоения–связности (7.5) обращалась бы в равенство? Возможно ли сформулировать эти ограничения в терминах графа расслоения–связности?

Задача 7.2 (10*). Обосновать гипотезу сепарабельности для линейных алгоритмов классификации.

8 Конъюнктивные логические закономерности

Мы добрались, наконец, до первого практического применения комбинаторной теории переобучения. А именно, оценки расслоения–связности будут применены для улучшения *логических методов классификации*.

§8.1 Логические методы классификации

Рассмотрим задачу классификации. Допустим, что каждому объекту $x_i \in \mathbb{X}$ соответствует *правильный ответ* $y_i \in \mathbb{Y}$, где $\mathbb{Y}$ — конечное множество имён классов. Объекты описываются набором n числовых *признаков* $f_j \colon \mathbb{X} \to \mathbb{R}$, $j = 1, \ldots, n$.

Логические методы классификации основаны на построении композиций информативных, хорошо интерпретируемых логических закономерностей.

Логические правила и требование интерпретируемости. *Предикатом* будем называть произвольное отображение вида $r \colon \mathbb{X} \to \{0, 1\}$. Если $r(x) = 1$, то будем говорить, что предикат *выделяет* объект x.

Правилом (rule) будем называть предикат из некоторого фиксированного семейства предикатов R. Правила отличаются от предикатов тем, что обладают свойством *интерпретируемости* — допускают запись на естественном языке в терминах предметной области, достаточно просты и понятны прикладным специалистам. Эти требования формализуются в самой конструкции семейства R. Мы рассмотрим только один вид правил, пожалуй, самый распространённый на практике — конъюнкции (логическое И) элементарных пороговых предикатов:

$$r(x;\theta) = \prod_{j \in J} \big[f_j(x) \leqslant_j \theta^j \big], \tag{8.1}$$

где $J \subseteq \{1, \ldots, n\}$ — подмножество признаков, $\theta^j \in \mathbb{R}$ — *порог* по j-му признаку, $\theta = (\theta^j)_{j \in J}$ — *вектор порогов*, $\leqslant_j$ — один из знаков отношения $\{\leqslant, \geqslant, =\}$. Число использованных в правиле признаков $|J|$ называется также *рангом конъюнкции*. Обычно оно ограничивается сверху в угоду той же интерпретируемости. Как утверждают психологи, людям трудно помнить и понимать правила, содержащие более 7 условий. На практике ограничение на $|J|$ устанавливается прикладными специалистами исходя из специфики задачи.

Примеры закономерностей. Из задачи медицинского прогнозирования: если возраст пациента выше 60 лет И ранее он перенёс инфаркт, то операцию делать не стоит. Из задачи кредитования физических лиц: если заёмщик указал в анкете свой домашний телефон И его зарплата превышает \$2000 в месяц И сумма кредита не превышает \$5 000, то кредит можно выдать. Из задачи распознавания спама: если в письме присутствует слово «бесплатно» И указан московский телефонный номер И домен отправителя находится в Китае, то это спам.

Понятия закономерности и информативности. *Закономерностью* класса $y \in \mathbb{Y}$ будем называть правило $r \in R$, выделяющее на заданной выборке $X \subseteq \mathbb{X}$ достаточно

много объектов класса y и мало объектов всех остальных классов. Для формализации этого требования вводят два функционала качества правил: $p(r, X)$ — число *положительных примеров* — объектов класса y, выделяемых правилом r, и $n(r, X)$ — число *отрицательных примеров* — объектов всех остальных классов, выделяемых правилом r. Для поиска закономерностей в семействе правил R по обучающей выборке X естественно ставить задачу двухкритериальной оптимизации:

$$p(r, X) = \#\big\{x_i \in X \;\big|\; r(x_i) = 1,\ y_i = y\big\} \to \max;$$
$$n(r, X) = \#\big\{x_i \in X \;\big|\; r(x_i) = 1,\ y_i \neq y\big\} \to \min.$$

На практике два функционала качества предпочитают сворачивать в один скалярный *критерий информативности* и решать задачу оптимизации $I(p, n) \to \max$. Известны десятки различных критериев информативности: эвристические свёртки типа $p - \gamma n$ [21], критерий бустинга $\sqrt{p} - \sqrt{n}$ [52], энтропийный критерий, индекс Джини, точный тест Фишера, тест χ^2, тест ω^2, и другие [46]. Многие критерии оценивают степень неслучайности разбиения выборки X на подмножества положительных и отрицательных примеров относительно исходного разбиения выборки X на классы. Однако ни один из критериев нельзя назвать наиболее обоснованным или безусловно предпочтительным. Выбор критерия информативности является эвристикой.

Обозначим через $P(X) = \#\{x_i \in X \colon y_i = y\}$ и $N(X) = \#\{x_i \in X \colon y_i \neq y\}$, соответственно, число положительных примеров (объектов фиксированного класса y) и число отрицательных примеров (объектов всех остальных классов) в выборке X.

Пример 8.1 (точный тест Фишера). Если принять в качестве нулевой гипотезы предположение, что предикаты $r(x)$ и $\big[y(x) = y\big]$ являются независимыми случайными величинами, то при фиксированном числе выделяемых значений $(p + n)$ число выделяемых отрицательных примеров n подчиняется гипергеометрическому распределению $h_{P+N}^{p+n,\,N}(n)$. Достигаемый уровень значимости даётся функцией гипергеометрического распределения. Он равен вероятности чисто случайной реализации наблюдаемых значений p и n. Чем он меньше, тем менее правдоподобна нулевая гипотеза. Поэтому он может служить мерой неслучайности величины $r(x)$, а неслучайность можно трактовать как закономерность. В качестве максимизируемого критерия информативности берут минус логарифм достигаемого уровня значимости:

$$I(p, n) = -\log H_{P+N}^{p+n,\,N}(n).$$

Пример 8.2 (энтропийный критерий). Асимптотическим приближением гипергеометрического критерия является *энтропийный критерий информативности* или выигрыш информации (information gain, IGain) [50]:

$$\mathrm{IGain}(p, n) = h\left(\frac{P}{P+N}\right) - \frac{p+n}{|X|}\, h\left(\frac{p}{p+n}\right) - \frac{|X|-p-n}{|X|}\, h\left(\frac{P-p}{|X|-p-n}\right),$$

где $h(q) = -q \log_2 q - (1-q)\log_2(1-q)$ — функция энтропии пары исходов с вероятностями q и $1-q$.

Пример 8.3 (критерий Джини). На практике используют также *индекс Джини* (Gini impurity), отличающийся от IGain только функцией $h(q) = 2q(1-q)$, которая на самом деле очень близка к функции энтропии [35].

Композиции закономерностей. Каждая закономерность выделяет лишь часть выборки и относит её к одному из классов. Поэтому закономерности можно рассматривать как «ущербные» классификаторы, а «полноценный» классификатор строить как композицию закономерностей. Одним из наиболее распространённых типов композиций является *взвешенное голосование*:

$$a(x) = \arg\max_{y\in\mathbb{Y}} \sum_{r\in R_y} w_r r(x),$$

где R_y — множество закономерностей класса y, $w_r \geqslant 0$ — вес закономерности r.

Построение таких композиций можно вести по-разному. Одна из стратегий заключается в том, чтобы сначала построить списки закономерностей R_y, затем, рассматривая закономерности $r(x)$ как новые признаки, построить на них линейный классификатор любым из известных методов, например, логистической регрессией или методом опорных векторов [39]. Другая стратегия заключается в том, чтобы строить закономерности по очереди, и для каждой закономерности r сразу определять вес w_r. Так, в частности, работают алгоритмы бустинга [37].

Переобученность логических закономерностей. Чтобы алгоритм классификации $a(x)$ имел высокую обобщающую способность, составляющие его закономерности $r(x)$ не должны быть переобучены. Недостаток стандартных критериев информативности в том, что они не учитывают переобучение, которое может возникать при оптимизации порогов θ^j. После оптимизации $p(r, X)$ и $n(r, X)$ по обучающей выборке X соответствующие величины $p' = p(r, \bar{X})$ и $n' = n(r, \bar{X})$ уже не будут оптимальными на контрольной выборке $\bar{X} = \mathbb{X}\backslash X$. Идея состоит в том, чтобы получить оценку расслоения–связности для вероятности переобучения конъюнктивных правил (8.1); затем обратить эту оценку и получить оценки величин p', n' на контроле через p, n на обучении; и, наконец, подставить оценки (p', n') в критерий информативности вместо (p, n). Таким образом, критерий отбора закономерностей изменится и будет, фактически, оценивать не конкретные правила, а наборы признаков J с учётом возможного переобучения порогов по каждому из признаков.

Минус предлагаемой схемы в том, что она не учитывает переобучение при выборе подмножеств J и весов w_r, значит, не даёт окончательного решения проблемы переобучения для алгоритма $a(x)$. Плюс же в том, что она ничего не меняет в методах поиска логических закономерностей, за исключением критерия информативности, поэтому её легко встраивать в имеющиеся библиотеки алгоритмов.

Начнём осуществление нашего замысла с описания множества векторов ошибок, индуцируемых конъюнкциями вида (8.1).

§8.2 Конъюнкции элементарных пороговых правил

Оценки вероятности переобучения, полученные в общем случае для алгоритмов, легко переносятся на логические правила, если определить индикатор ошибки как $I(r, x_i) = \big[r(x_i) \neq [y_i{=}y]\big]$, где $y \in \mathbb{Y}$ — фиксированный класс. После этого для правил, как и ранее для алгоритмов, вводятся векторы ошибок $\vec{r} = \big(I(r, x_i)\big)_{i=1}^{L}$, отношения порядка и предшествования.

Будем полагать, что в правилах вида (8.1) множество признаков J фиксировано и все знаки сравнения суть $\leqslant$. Предполагается также, что перебор $2^{|J|}C_n^{|J|}$ сочетаний наборов признаков и знаков $\leqslant, \geqslant$, не обязательно полный, осуществляется некоторым стандартным механизмом поиска информативных правил. Наша задача заключается в том, чтобы оценить вероятность переобучения, возникающего в результате оптимизации *вектора порогов* $\theta = (\theta^j)_{j\in J}$ по обучающей выборке X.

Описание структуры классов эквивалентности в терминах стандартных представителей и граничных подмножеств, которое мы сейчас рассмотрим, предложено А. Ивахненко. Им же подготовлены все иллюстрации, выполнена программная реализация и эксперименты на реальных задачах классификации.

Допустим, что значения $x_i^j = f_j(x_i)$ каждого признака $j \in J$ на всех объектах $x_i \in \mathbb{X}$ попарно различны. Тогда без ограничения общности можно предполагать, что все признаки принимают целые значения $1, \ldots, L$, и никакие два объекта не имеют равных значений одного признака. Значения порогов θ^j в правилах (8.1) можно выбирать также из целых значений $0, \ldots, L$.

Индикатор ошибки индуцирует на множестве правил R *классы эквивалентности*. Два правила эквивалентны, $r \sim r'$, если их векторы ошибок совпадают.

Для произвольных векторов $u = (u^j)_{j\in J}$, $v = (v^j)_{j\in J}$ введём естественное отношение порядка: $(u \leqslant v) \;\leftrightarrow\; \forall j \in J\ (u^j \leqslant v^j)$. Положим $(u < v) \;\leftrightarrow\; (u \leqslant v$ и $u \neq v)$.

Пример 8.4. На рис. 8.1 показан пример задачи с $n = 2$ признаками, $L = 10$ объектами и семейство правил $r(x;\theta) = [x \leqslant \theta] = [x^1 \leqslant \theta^1]\,[x^2 \leqslant \theta^2]$. Каждому правилу соответствует узел целочисленной прямоугольной сетки $(\theta^1, \theta^2) \in \{0, \ldots, L\}^2$.

Лемма 8.1. *Пусть $E \subseteq R$ — класс эквивалентности правил, θ_r^j — порог по j-му признаку в правиле r. Тогда классу E принадлежит также правило $r(x;\theta_E)$, где*

$$\theta_E^j = \min_{r\in E} \theta_r^j, \quad j \in J.$$

Доказательство. В силу эквивалентности и бинарности правил $r \in E$, предикат

$$r_E(x) = \prod_{r\in E} r(x;\theta_r)$$

принимает на всех объектах $x \in \mathbb{X}$ те же значения, что и любое правило r из E, $r_E(x) = r(x;\theta_r)$. Кроме того, предикат r_E представим в виде (8.1):

$$r_E(x) = \prod_{r\in E}\prod_{j\in J}[x^j \leqslant \theta_r^j] = \prod_{j\in J}[x^j \leqslant \min_{r\in E}\theta_r^j] = \prod_{j\in J}[x^j \leqslant \theta_E^j] = r(x;\theta_E).$$

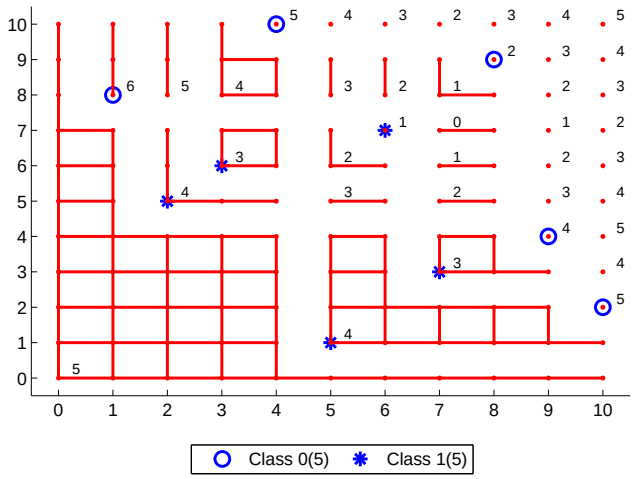


Рис. 8.1. Двумерная выборка из $L = 10$ объектов, по 5 объектов в каждом классе. Объекты отмечены крупными точками. Правилам соответствуют мелкие точки в узлах сетки. Эквивалентные правила соединены отрезками. Рядом с каждым классом эквивалентности указано число ошибок на генеральной выборке $n(r, \mathbb{X})$.

Таким образом, правило $r(x;\theta_E)$ также принадлежит E. Лемма доказана. ■

Будем называть правило $r_E(x) \equiv r(x;\theta_E)$ *стандартным представителем* класса эквивалентности E. На рис. 8.1 стандартные представители соответствуют левым нижним точкам каждого класса эквивалентности: $(0,0)$, $(1,8)$, $(2,5)$, $(5,1)$, и т. д.

Граничной точкой подмножества $S \subseteq \mathbb{X}$ назовём вектор θ_S с координатами

$$\theta_S^j = \max_{x \in S} x^j, \ j \in J.$$

Назовём объект x подмножества S *граничным*, если $x^j = \theta_S^j$ при некотором j.
Назовём подмножество $S \subseteq \mathbb{X}$ *граничным*, если все его объекты граничны.
Пустое множество будем считать граничным с граничной точкой $\theta_\varnothing^j = 0$, $j \in J$.
Заметим, что $r(x, \theta_S) = 1$ для любого $x \in S$.

Лемма 8.2. *Если вектор порогов θ является граничной точкой некоторого граничного подмножества, то это подмножество однозначно определяется по вектору θ:*

$$S_\theta = \bigcup_{j \in J} \big\{ x \in \mathbb{X} \;\big|\; x^j = \theta^j, \ r(x,\theta) = 1 \big\}. \tag{8.2}$$

Доказательство. Из того, что θ является граничной точкой, следует, что либо $\theta = \theta_\varnothing$, и тогда $S_\theta = \varnothing$, либо для каждого индекса $j \in J$ найдётся объект $x \in \mathbb{X}$ такой, что $x^j = \theta^j$ и $x^{\bar{\jmath}} \leqslant \theta^{\bar{\jmath}}$ для всех индексов $\bar{\jmath} \in J \setminus \{j\}$. Тогда $r(x,\theta) = 1$, и из (8.2) следует, что объект x лежит в S_θ. В силу произвольности индекса $j \in J$ вектор θ является граничной точкой подмножества S_θ. Подмножество S_θ является граничным, поскольку каждый его объект, согласно (8.2), является граничным. ■

Замечание 8.1. Каждое множество под знаком объединения в (8.2) содержит не более одного объекта благодаря предположению, что значения каждого признака попарно различны на всей выборке $\mathbb{X}$. Позже мы откажемся от этого предположения, но при этом придётся усложнить определение граничного подмножества.

Теорема 8.3. *Каждому классу эквивалентности E взаимно однозначно соответствует граничное подмножество S, такое, что $\theta_E = \theta_S$.*

Доказательство. Рассмотрим произвольный класс эквивалентности E со стандартным представителем $r(x;\theta_E)$. В силу Леммы 8.1 уменьшение порога θ_E^j на единицу по любому признаку $j \in J$ приводит к изменению значения $r(x;\theta_E)$ на некотором объекте $x \in \mathbb{X}$. Это может быть только объект, лежащий ниже, $x \leqslant \theta_E$, причём значение $r(x,\theta_E)$ может только уменьшиться: $[x^j \leqslant \theta_E^j] = 1$, $[x^j \leqslant \theta_E^j - 1] = 0$. Отсюда следует равенство $x^j = \theta_E^j$. Значит, x является граничным объектом множества $\{x' \in \mathbb{X}\colon x' \leqslant \theta_E\}$. В силу произвольности j это означает, что вектор θ_E является граничной точкой и, согласно лемме 8.2, задаёт граничное подмножество.

Верно и обратное. Произвольному граничному подмножеству S соответствует граничная точка θ_S. Правило $r(x;\theta_S)$ лежит в некотором классе эквивалентности E и является его стандартным представителем, поскольку уменьшение порога θ_S^j по любой из координат приведёт к изменению значения $r(x;\theta_S)$ на одном из граничных объектов множества S.

Таким образом, существует взаимно однозначное соответствие между граничными подмножествами S и стандартными представителями классов эквивалентности E, причём $\theta_E = \theta_S$. Теорема доказана. ■

Свойства граничных подмножеств. Обозначим через M_q множество всех граничных подмножеств мощности q и перечислим их основные свойства.

1. M_1 состоит из всех L одноэлементных подмножеств $\{x\} \subset \mathbb{X}$.
2. M_2 состоит из всех пар несравнимых объектов из $\mathbb{X}$.
3. Мощность граничных подмножеств не может превышать ранга конъюнкции: $M_q = \varnothing$ при $q > |J|$.
4. Граничное подмножество может состоять только из попарно несравнимых объектов. Однако не всякое подмножество из трёх и более попарно несравнимых объектов является граничным.
5. Любое подмножество $S' \subset S$ граничного подмножества S также граничное.

Исходя из этих свойств, можно предложить следующий алгоритм перебора всех граничных подмножеств.

На первом шаге строится множество M_1 всех L одноэлементных подмножеств. Далее на каждом шаге $q = 2, \ldots, |J|$ к каждому подмножеству $S' \in M_{q-1}$ добавляется

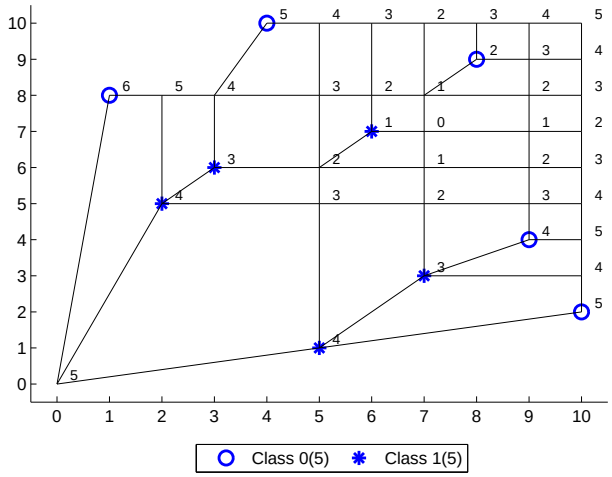


Рис. 8.2. Граф связей между стандартными представителями классов эквивалентных правил для выборки, представленной на Рис. 8.1.

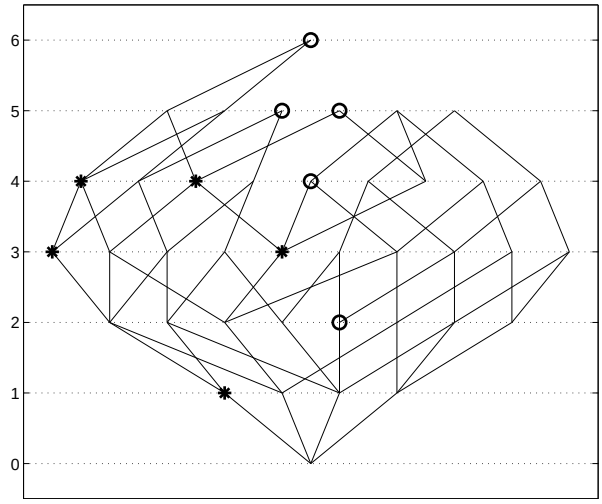


Рис. 8.3. Граф расслоения–связности, изоморфный графу связей на Рис. 8.2. По вертикальной оси отложено число ошибок правил $n(r, \mathbb{X})$.

всеми возможными способами ещё один объект $x \in \mathbb{X} \setminus S'$, и если полученное подмножество $S = S' \cup \{x\}$ граничное (что легко проверяется по определению), то оно включается в M_q.

Для вычисления оценки расслоения–связности (7.5) этот алгоритм не достаточно эффективен, так как он не перебирает правила по слоям снизу вверх и не подсчитывает характеристики связности и неоптимальности каждого правила.

Графические представления классов эквивалентности. На рис. 8.2 изображен *граф связей* между классами эквивалентности. Вершины графа соответствуют стандартным представителям классов эквивалентности. Рёбрами соединены правила, векторы ошибок которых различаются на одном объекте. Граф связей зависит только от объектов x_i, но не зависит от их классификаций y_i.

Если учесть классификации объектов, то для каждого класса эквивалентности E можно вычислить число ошибок $n(r_E, \mathbb{X})$. На Рис. 8.2 оно указано рядом с каждой точкой — стандартным представителем класса эквивалентности. Граф расслоения-связности данного семейства правил изоморфен графу связей и является многодольным. Его m-й слой образуется всеми правилами, для которых $n(r_E, \mathbb{X}) = m$, см. Рис. 8.3. Слои располагаются в порядке возрастания числа ошибок снизу вверх.

§8.3 Применение оценки расслоения–связности

Рассмотрим теперь алгоритм перебора правил, позволяющий эффективно вычислять характеристики связности и неоптимальности каждого правила и, двигаясь по слоям снизу вверх, вовремя прерывать вычисление оценки расслоения–связности.

Построение окрестности правила. Рассмотрим следующую вспомогательную задачу. Задано правило $r(x;\theta)$ с вектором порогов $\theta = (\theta^j)_{j\in J}$. Требуется построить его *окрестность* V_θ — множество всех правил $r(x,\theta')$, векторы ошибок которых отличаются от его вектора ошибок только на одном объекте.

Окрестность V_θ соответствует подграфу графа расслоения–связности, состоящему из вершины θ, всех соседних с ней вершин и инцидентных ей рёбер. При построении окрестности будут заодно формироваться и характеристики правила $r(x,\theta)$, необходимые для вычисления оценки расслоения–связности:

X_θ — порождающее множество, $q(\theta) = |X_\theta|$ — верхняя связность;

X'_θ — запрещающее множество, $r(\theta) = |X'_\theta|$ — неоптимальность;

D_θ — множество объектов входящих рёбер, $d(\theta) = |D_\theta|$ — нижняя связность;

$m(\theta)$ — число ошибок правила на генеральной выборке.

Будем строить только такие векторы порогов θ', которые являются граничными точками и, согласно Теореме 8.3, одновременно стандартными представителями классов эквивалентности правил. Построение разбивается на два этапа, см. Алгоритм 8.1.

На первом этапе (шаги 1–5) строятся все соседние правила $r(x;\theta')$, получаемые из θ путём уменьшения некоторых порогов, $\theta' \leqslant \theta$. Для этого из граничного множества S_θ поочерёдно исключается один из объектов и соответствующие пороги θ^j, по которым он и был граничным, уменьшаются. Число получаемых таким способом соседних правил в точности равно $|S_\theta|$.

На втором этапе (шаги 6–11) строятся все соседние правила $r(x;\theta')$, получаемые путём увеличения некоторых порогов, $\theta' \geqslant \theta$. Это более сложный случай, и здесь приходится прибегать к рекурсивной процедуре. Сначала делается подготовительная работа (шаги 6, 7): по каждой координате $j \in J$ определяется максимальная граница $\bar\theta^j$, выше которой соседних правил быть не может. Это необязательное построение, но оно позволит в дальнейшем сократить поиск. Каждый объект выборки $x \in \mathbb{X}$ может находиться в одном из трёх состояний: $x.\mathsf{проверен} \in \{\mathsf{нет}, \mathsf{плохой}, \mathsf{хороший}\}$. Сначала все объекты не проверены. Затем просматриваются все объекты $x \in \mathbb{X}$, лежащие ниже максимальной границы $\bar\theta$, но не покрываемые правилом $r(x;\theta)$. Это означает, что хотя бы по одной координате объект x лежит выше порога: $\theta^j < x^j$. Для каждого такого объекта вызывается процедура $\mathsf{Проверить}(x)$. Она устанавливает состояние

Алгоритм 8.1. Построение окрестности V_θ правила $r(x;\theta)$.

Вход:

J — набор признаков, $\theta = (\theta^j)_{j\in J}$ — правило, y — класс правил, $\mathbb{X}$ — выборка.

Выход:

V_θ, X_θ, X'_θ, D_θ, $q(\theta)$, $r(\theta)$, $d(\theta)$, $m(\theta)$.

Этап 1 — построение окрестных правил путём уменьшения порогов:

$V_\theta := \varnothing$;
для всех $x \in S_\theta$
  для всех $j \in J$ таких, что $x^j = \theta^j$
    $\theta'^j := \max\{x_i^j \mid x_i \in \mathbb{X},\ x_i < \theta\}$;
  Добавить(θ, θ', x);

Подготовка к этапу 2 — поиск максимальной граничной точки $\bar\theta$:

для всех $j \in J$
  $\bar\theta^j := \max\{L, x^j \mid x \in \mathbb{X},\ x^j > \theta^j,\ x^t \leqslant \theta^t,\ t \neq j\}$;

Этап 2 — построение окрестных правил путём увеличения порогов:

для всех $j \in J$
  для всех x таких, что $\theta^j < x^j \leqslant \bar\theta^j$
    если $x < \bar\theta$ и x.проверен = нет **то**
      Проверить(x);

ПРОЦЕДУРА Проверить(x)
для всех $j \in J$ таких, что $\theta^j < x^j$
  для всех $\tilde x$ таких, что $\theta^j < \tilde x^j < x^j$
    если $\theta < \tilde x < x$ **то**
      x.проверен := плохой;
      если $\tilde x$.проверен = нет **то** Проверить$(\tilde x)$;
      выход;
x.проверен := хороший;
$\theta'^j := \max\{\theta^j, x^j\}$, для всех $j \in J$;
Добавить(θ, θ', x);

ПРОЦЕДУРА Добавить(θ, θ', x_i)
добавить θ' в список V_θ;
если $r(x_i;\theta) = [y_i{=}y]$ **то**
  правило θ' находится слоем выше, чем θ:
  $X_\theta := X_\theta \cup \{x_i\}$; $q(\theta) := |X_\theta|$;
иначе
  правило θ' находится слоем ниже, чем θ:
  $D_\theta := D_\theta \cup \{x_i\}$; $d(\theta) := |D_\theta|$;
  $X'_\theta := X'_\theta \cup X'_{\theta'} \cup \{x_i\}$; $r(\theta) := |X'_\theta|$;
  $m(\theta) := m(\theta') + 1$;

Алгоритм 8.2. Вычисление оценки расслоения–связности для семейства пороговых конъюнкций.

Вход: J — набор признаков, y — класс правил, $\mathbb{X}$ — выборка.
Выход: Q_ε — оценка вероятности переобучения (7.5).

1: $\Theta := \operatorname{Arg}\min\limits_{\theta} n(\theta, \mathbb{X})$;
2: $m(\theta) := n(\theta, \mathbb{X})$, для всех $\theta \in \Theta$;
3: $Q_\varepsilon := 0$;
4: **повторять**
5: $\quad Q_{\varepsilon,m} := 0; \quad \Theta' := \varnothing$;
6: $\quad$ **для всех** $\theta \in \Theta$
7: $\quad\quad$ построить окрестность V_θ с помощью Алгоритма 8.1;
8: $\quad\quad Q_{\varepsilon,m} := Q_{\varepsilon,m} + \frac{1}{C_L^\ell} C_{L-q(\theta)-r(\theta)}^{\ell-q(\theta)} H_{L-q(\theta)-r(\theta)}^{\ell-q(\theta),\, m(\theta)-r(\theta)} \left(\frac{\ell}{L}(m(\theta) - \varepsilon k)\right)$;
9: $\quad\quad \Theta' := \Theta' \cup \left\{\theta' \in V_\theta \colon m(\theta') = m(\theta) + 1\right\}$;
10: $\quad\quad Q_\varepsilon := Q_\varepsilon + Q_{\varepsilon,m}; \quad \Theta := \Theta'$;
11: **пока** вклад слоя $Q_{\varepsilon,m}$ не станет мал.

объекта x.хороший, если существует правило $r(x;\theta')$, покрывающее только объект x и все объекты, покрываемые правилом $r(x;\theta)$. Если же правило $r(x;\theta')$, наряду с x, покрывает ещё один объект $\tilde{x}$, не покрываемый правилом $r(x;\theta)$, то устанавливается состояние объекта x.плохой, и процедура Проверить$(\tilde{x})$ вызывается рекурсивно для объекта $\tilde{x}$. Каждый «хороший» объект x индуцирует соседнее правило $\theta' = \max\{\theta, x\}$, которое и добавляется в список V_θ. Введение статусов объектов позволяет избежать их повторного перебора в основном цикле второго этапа.

Этими двумя этапами поиск окрестных правил ограничивается. Другие случаи рассматривать не нужно, так как если порог уменьшается по одной координате, $\theta'^j < \theta^j$, и увеличивается по другой, $\theta'^t > \theta^t$, то граничные точки θ и θ' не могут быть соседними, поскольку векторы ошибок соответствующих им правил отличаются, как минимум, на двух объектах.

Эффективная реализация Алгоритма 8.1 предполагает, что по каждому признаку f_j заранее строится индекс — массив номеров объектов, упорядоченных по возрастанию значений признака. Тогда на шагах 4, 7, 9, 14 можно будет просматривать только нужные объекты, не перебирая всю выборку.

Послойный перебор классов эквивалентности. Алгоритм 8.2 вычисляет Q_ε — оценку вероятности переобучения (7.5), перебирая правила по слоям снизу вверх. Сначала формируется нижний слой. На каждом шаге слой Θ состоит из правил θ с одинаковым числом ошибок $m(\theta)$ на генеральной выборке. Для каждого правила θ вычисляется его вклад в вероятность переобучения, строится окрестность, и объединение верхних частей всех полученных окрестностей становится следующим слоем Θ'. Шаги повторяются до тех пор, пока не будет набрано достаточное число слоёв и вклад очередного слоя не окажется пренебрежимо мал.

Резюме

Логические алгоритмы классификации представляют собой композиции закономерностей — информативных, достаточно простых, хорошо интерпретируемых логических правил. На практике в качестве правил чаще всего используются конъюнкции пороговых условий над числовыми признаками. Данное семейство правил описывается системой всех граничных подмножеств мощности, не превышающей ранга конъюнкции. Для практического вычисления оценки расслоения–связности предлагается перебирать правила по слоям снизу вверх, и прекращать перебор в тот момент, когда новые слои перестанут вносить существенный вклад в оценку.

В следующей лекции мы сделаем небольшой экскурс в математическую статистику и рассмотрим задачу оценивания эмпирических распределений. Заодно познакомимся с такими интересными математическими объектами, как усечённый треугольник Паскаля и одномерные случайные блуждания. Они понадобятся нам через лекцию, когда мы вернёмся к функционалу равномерного отклонения и покажем, что связность для него учесть можно, а расслоение — нельзя. Затем мы продолжим изучение модельных семейств алгоритмов и покажем, что в некоторых нетривиальных случаях оценки расслоения–связности могут быть точными.

Упражнения

Задача 8.1 (1). Доказать, что не всякое подмножество из трёх и более попарно несравнимых объектов является граничным.

Задача 8.2 (2). Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для семейства правил

$$r(x;\theta_1,\theta_2) = \big[\theta_1 \leqslant x \leqslant \theta_2\big],$$

где все значения $x_i \in \mathbb{X} \subset \mathbb{R}$ попарно различны. Подсказка: рассмотреть эквивалентное семейство двухпризнаковых конъюнкций $r(x;\theta_1,\theta_2) = \big[-x \leqslant \theta_1\big]\big[x \leqslant \theta_2\big]$.

Задача 8.3 (5*). Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для случая, когда индикатор ошибки определяется отдельно для положительных и отрицательных примеров:

$$I_p(r,x_i) = \big[r(x_i)=0\big]\big[y_i = y\big];$$
$$I_n(r,x_i) = \big[r(x_i)=1\big]\big[y_i \neq y\big].$$

Задача 8.4 (5). Описать структуру классов эквивалентности и предложить эффективный алгоритм их послойного перебора для случая, когда каждый признак может принимать одинаковые значения на различных объектах.

Задача 8.5 (2). Обобщить Алгоритм 8.2 на тот случай, когда на шаге 1 вместо множества всех правил, доставляющих глобальный минимум числу ошибок, находится

лишь одно локально оптимальное правило. Предусмотреть возможность переходов от правил m-го слоя к правилам не только $(m+1)$-го слоя, но и $(m-1)$-го слоя.

9 Оценивание эмпирического распределения и случайное блуждание

В Лекции 2 мы оценивали вероятность большого отклонения частот в двух выборках. В слабой вероятностной аксиоматике эта задача является естественным аналогом закона больших чисел и имеет точное решение. Оценивание вероятности большого отклонения двух функций распределения — это ещё одна классическая задача математической статистики. Её можно интерпретировать двумя способами. Во-первых, как задачу предсказания: дана эмпирическая функция распределения случайной величины на наблюдаемой выборке; требуется оценить её эмпирическую функцию распределения на скрытой выборке. Во-вторых, как задачу проверки гипотезы однородности: даны две наблюдаемые выборки; требуется определить, получены ли они из одного распределения. Для её решения используются различные статистические критерии, в частности, двухвыборочный критерий Смирнова.

Мы рассмотрим постановку и точное решение этих классических задач в слабой аксиоматике и покажем, что они тесно связаны с такими математическими объектами, как усечённый треугольник Паскаля и одномерные случайные блуждания.

§9.1 Эмпирическое распределение

Определим для произвольной функции $\xi\colon \mathbb{X} \to \mathbb{R}$ и произвольной конечной выборки $U \subseteq \mathbb{X}$ *эмпирическую функцию распределения* $F_\xi\colon \mathbb{R} \to [0,1]$ как долю объектов x выборки U, для которых значение $\xi(x)$ не превосходит z:

$$F_\xi(z, U) = \frac{1}{|U|} \sum_{x \in U} [\xi(x) \leqslant z].$$

Определим одностороннее и двустороннее *равномерное отклонение эмпирических функций распределения*:

$$D^+(X) = \max_{z \in \mathbb{R}} \big(F_\xi(z, \bar{X}) - F_\xi(z, X)\big);$$
$$D^-(X) = \max_{z \in \mathbb{R}} \big(F_\xi(z, X) - F_\xi(z, \bar{X})\big);$$
$$D(X) = \max_{z \in \mathbb{R}} \big|F_\xi(z, \bar{X}) - F_\xi(z, X)\big| = \max\big\{D^-(X), D^+(X)\big\}.$$

Задача состоит в том, чтобы найти вероятность большого отклонения эмпирических функций распределения:

$$\mathsf{P}\big[D(X) > \varepsilon\big] \leqslant \eta(\varepsilon), \tag{9.1}$$

где вместо $D(X)$ могут быть также подставлены $D^-(X)$ или $D^+(X)$.

В сильной аксиоматике имеет место следующий факт [29].

Теорема 9.1 (Н. В. Смирнов). *Если $X, \bar{X} \subseteq \mathscr{X}$ — случайные, независимые, одинаково распределённые выборки; $\xi\colon \mathscr{X} \to \mathbb{R}$ — случайная величина с непрерывным распределением, то справедливы асимптотические оценки*

$$\lim_{\ell,k\to\infty} \mathsf{P}\big\{D^{\pm}(X) \geqslant \varepsilon\big\} = \exp\left(-2\varepsilon^2 \tfrac{\ell k}{\ell+k}\right); \tag{9.2}$$

$$\lim_{\ell,k\to\infty} \mathsf{P}\big\{D(X) \geqslant \varepsilon\big\} = 2\sum_{i=1}^{\infty}(-1)^{i+1}\exp\left(-2\varepsilon^2 \tfrac{\ell k}{\ell+k} i^2\right); \tag{9.3}$$

Правая часть (9.3) представима также в виде $1-K\left(\varepsilon\sqrt{\frac{\ell k}{\ell+k}}\right)$, где $K(z)$ — *функция распределения Колмогорова*:

$$K(z) = \sum_{i=-\infty}^{\infty}(-1)^i e^{-2z^2i^2}.$$

Известны и неасимптотические точные оценки, но они имеют достаточно громоздкий вид [12]. Мы покажем, что точные оценки могут быть выражены более элегантно через усечённый треугольник Паскаля [32], причём доказательство имеет прозрачный геометрический смысл.

§9.2 Усечённый треугольник Паскаля

Пусть $g_m^-,\ g_m^+,\ m = 0,\ldots,L$ — две неубывающие последовательности, удовлетворяющие условию $0 \leqslant g_m^- \leqslant g_m^+ \leqslant m$.

Определение 9.1. *Усечённым треугольником Паскаля с нижней границей g_m^- и верхней границей g_m^+ называется целочисленная функция $G_m^s = G_m^s[g_m^-, g_m^+]$, определяемая рекуррентными соотношениями*

$$\begin{aligned} &G_0^s = [s=0], \quad s \in \mathbb{Z};\\ &G_m^s = \big(G_{m-1}^s + G_{m-1}^{s-1}\big)\big[g_m^- \leqslant s \leqslant g_m^+\big], \quad m \in \mathbb{N}, \quad s \in \mathbb{Z}. \end{aligned} \tag{9.4}$$

Усечённый треугольник Паскаля вычисляется по тому же рекуррентному правилу, что и классический треугольник Паскаля, если в нём обнулить все элементы, лежащие за пределами границ $[g_m^-, g_m^+]$. «Неусечённый» треугольник Паскаля $G_m^s[0,m]$ совпадает с классическим и даёт биномиальные коэффициенты C_m^s.

При начальном условии $G_0^s = [s=0]$ ненулевыми могут быть только элементы G_m^s при $0 \leqslant s \leqslant m$. Другие начальные условия приводят к различным неклассическим обобщениям треугольника Паскаля, но мы их не будем рассматривать.

Определим для произвольных $\varepsilon > 0$ и $m = 0, 1, 2, \ldots$, линейные границы (в дальнейшем аргумент ε иногда будем опускать):

$$\begin{aligned} g_m^+(\varepsilon) &= \tfrac{\ell}{L}(m + \varepsilon k);\\ g_m^-(\varepsilon) &= \tfrac{\ell}{L}(m - \varepsilon k). \end{aligned}$$

На рис. 9.2 и 9.4 приведены примеры четырёх вариантов усечения треугольника Паскаля с такими границами. В отличие от привычного способа изображения, треугольники «положены на бок» путём поворота на 90° против часовой стрелки.

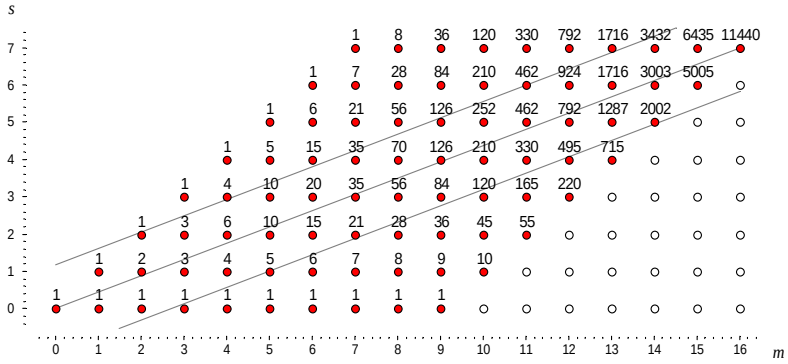


Рис. 9.1. Классический треугольник Паскаля $C_m^s = G_m^s[0, m]$ при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

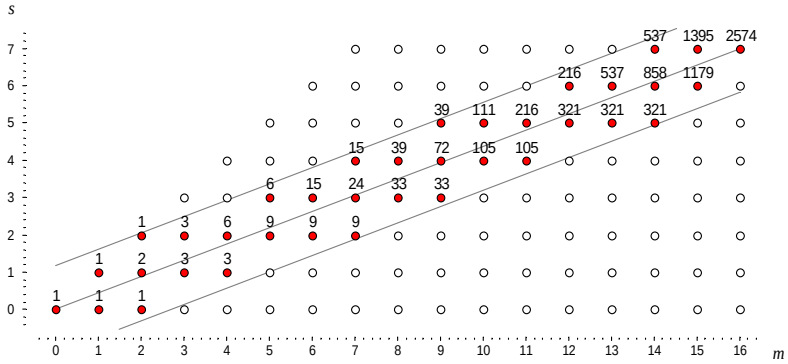


Рис. 9.2. Усечённый треугольник Паскаля $G_m^s\big[g_m^-(\varepsilon), g_m^+(\varepsilon)\big]$ при $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

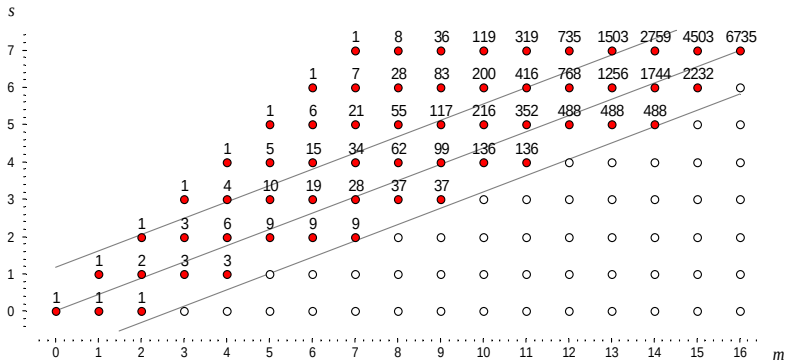


Рис. 9.3. Усечённый слева треугольник Паскаля $G_m^s\big[g_m^-(\varepsilon), m\big]$, $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

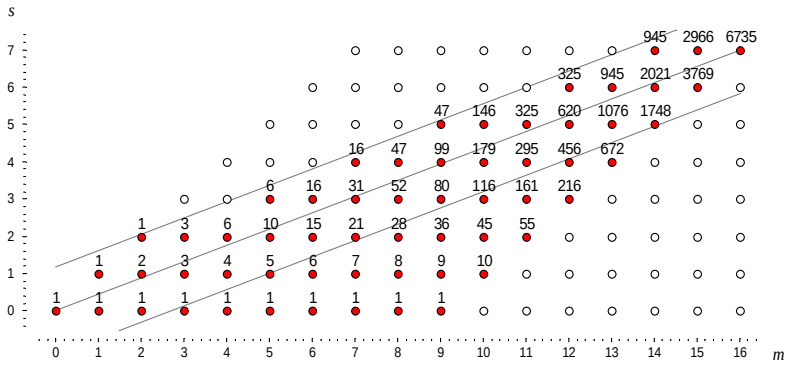


Рис. 9.4. Усечённый справа треугольник Паскаля $G_m^s\big[0, g_m^+(\varepsilon)\big]$, $L = 16$, $\ell = 7$, $\varepsilon = 0.3$.

§9.3 Теорема Смирнова

Теорема 9.2. *Для произвольной конечной выборки $\mathbb{X}$ и произвольной функции $\xi\colon \mathbb{X} \to \mathbb{R}$, значения которой попарно различны на элементах выборки $\mathbb{X}$, справедливы точные оценки:*

$$\mathsf{P}\big[D^+(X) \leqslant \varepsilon\big] = G_L^\ell\big[0, g_L^+(\varepsilon)\big]/C_L^\ell; \tag{9.5}$$

$$\mathsf{P}\big[D^-(X) \leqslant \varepsilon\big] = G_L^\ell\big[g_L^-(\varepsilon), L\big]/C_L^\ell; \tag{9.6}$$

$$\mathsf{P}\big[D(X) \leqslant \varepsilon\big] = G_L^\ell\big[g_L^-(\varepsilon), g_L^+(\varepsilon)\big]/C_L^\ell. \tag{9.7}$$

Доказательство. 1. Составим вариационный ряд значений функции $\xi(x)$ на элементах выборки: $\xi(x^{(1)}) < \xi(x^{(2)}) < \cdots < \xi(x^{(L)})$. Здесь все неравенства строгие в силу условия попарной различности.

Обозначим $b_i = b_i(X) = [x^{(i)} \in X]$. Бинарная последовательность $b_1, \ldots, b_L$ содержит ровно ℓ единиц и k нулей.

Воспользуемся определением функции распределения:

$$\begin{aligned} D(X) &= \max_{z\in\mathbb{R}} \big|F_\xi(z, \bar{X}) - F_\xi(z, X)\big| = \\ &= \max_{z\in\mathbb{R}} \left|\frac{1}{k}\sum_{i=1}^{L}[x_i \in \bar{X}][\xi(x_i) < z] - \frac{1}{\ell}\sum_{i=1}^{L}[x_i \in X][\xi(x_i) < z]\right|. \end{aligned} \tag{9.8}$$

Изменим порядок слагаемых в суммах, теперь суммируя их в порядке возрастания значений $\xi(x_i)$. Это равносильно тому, что в данной формуле все вхождения x_i заменятся на $x^{(i)}$. Тогда можно убрать сомножитель $\big[\xi(x^{(i)}) < z\big]$, заменив верхний предел суммирования на $m = \max\big\{i\colon \xi(x^{(i)}) < z\big\}$, и максимум брать не по действительному параметру z, а по целочисленному параметру m:

$$\begin{aligned} D(X) &= \max_{m=1..L} \left|\frac{1}{k}\sum_{i=1}^{m}\underbrace{[x^{(i)} \in \bar{X}]}_{1-b_i} - \frac{1}{\ell}\sum_{i=1}^{m}\underbrace{[x^{(i)} \in X]}_{b_i}\right| = \\ &= \max_{m=1..L}\left|\frac{m}{k} - \frac{\ell+k}{\ell k}\sum_{i=1}^{m} b_i\right| = \frac{L}{\ell k}\max_{m=1..L}\left|B_m - \frac{m\ell}{L}\right|, \end{aligned}$$

где $B_m = B_m(X) = b_1 + \cdots + b_m$.

Таким образом, равномерное отклонение эмпирических распределений на выборках X и $\bar{X}$ выражается через равномерное отклонение числа единиц в первых m членах последовательности $b_1, \ldots, b_L$ от «ожидаемого» числа единиц $m\ell/L$.

Теперь запишем долю разбиений выборки $\mathbb{X}$, при которых равномерное отклонение эмпирических распределений не превышает пороговую точность ε:

$$\begin{aligned}
&\mathsf{P}\big[D(X) \leqslant \varepsilon\big] = \\
&= \mathsf{P}\left[\max_{m=1..L} \left|B_m - \frac{m\ell}{L}\right| \leqslant \frac{\varepsilon\ell k}{L}\right] = \\
&= \mathsf{P}\left[\max_{m=1..L} \Big(-B_m + \underbrace{\big(\tfrac{m\ell}{L} - \tfrac{\varepsilon\ell k}{L}\big)}_{g_m^-(\varepsilon)}\Big) \leqslant 0\right]\left[\max_{m=1..L} \Big(B_m - \underbrace{\big(\tfrac{m\ell}{L} + \tfrac{\varepsilon\ell k}{L}\big)}_{g_m^+(\varepsilon)}\Big) \leqslant 0\right] = \\
&= \frac{1}{C_L^\ell} \sum_{(X,\bar{X})} \prod_{m=1}^{L} \big[g_m^-(\varepsilon) \leqslant B_m(X) \leqslant g_m^+(\varepsilon)\big]. \qquad (9.9)
\end{aligned}$$

Последнее равенство следует из тождества $\big[\max\limits_m A_m \leqslant 0\big] = \prod\limits_m \big[A_m \leqslant 0\big]$.

2. Рассмотрим подвыборку $X^m = \{x^{(1)}, \ldots, x^{(m)}\}$, состоящую из первых m членов вариационного ряда. Возьмём максимальное (по включению) подмножество N разбиений $(X, \bar{X})$, удовлетворяющих двум условиям:

1) они индуцируют попарно различные разбиения подвыборки X^m;

2) ровно s объектов из X^m попадают в X.

Очевидно, число этих разбиений $|N| = C_m^s$. Представим множество разбиений N в виде объединения непересекающихся подмножеств $N_0 = \{(X, \bar{X}) \in N \colon b_m(X) = 0\}$ и $N_1 = \{(X, \bar{X}) \in N \colon b_m(X) = 1\}$. Очевидно, $|N_0| = C_{m-1}^s$, $|N_1| = C_{m-1}^{s-1}$.

Нас будет интересовать выражение $H_m^s = \sum\limits_{(X,\bar{X})} \prod\limits_{r=1}^{m} \big[g_r^- \leqslant B_r(X) \leqslant g_r^+\big]$, поскольку правая часть (9.9) равна H_L^ℓ / C_L^ℓ. Разобьём в этом выражении сумму по N на две суммы — по N_0 и по N_1, и ещё заметим, что $B_m(X) = s$ для всех $(X, \bar{X}) \in N$:

$$\begin{aligned}
H_m^s &= \underbrace{\sum_{(X,\bar{X})\in N_0} \prod_{r=1}^{m-1} \big[g_r^- \leqslant B_r(X) \leqslant g_r^+\big]}_{H_{m-1}^s} \big[g_m^- \leqslant s \leqslant g_m^+\big] + \\
&+ \underbrace{\sum_{(X,\bar{X})\in N_1} \prod_{r=1}^{m-1} \big[g_r^- \leqslant B_r(X) \leqslant g_r^+\big]}_{H_{m-1}^{s-1}} \big[g_m^- \leqslant s \leqslant g_m^+\big] = \\
&= \big(H_{m-1}^s + H_{m-1}^{s-1}\big)\big[g_m^- \leqslant s \leqslant g_m^+\big].
\end{aligned}$$

Таким образом, получена рекуррентная формула для H_m^s, формально совпадающая с формулой усечённого треугольника Паскаля (9.4). Осталось только проверить граничные случаи.

При $m = 1$ и фиксированном $s \in \{0, 1\}$ имеется только одно разбиение, $|N| = 1$, следовательно, $H_1^s = \big[g_m^- \leqslant s \leqslant g_m^+\big]$, что совпадает с (9.4).

При $s = 0$ и произвольном $m = 1, \ldots, k$ имеется только одно разбиение, $|N| = 1$, причём ни один объект из X^m не попадает в X. Это означает, что $B_r = 0$ при всех $r = 1, \ldots, m$. Но тогда $H_m^0 = \prod_{r=1}^m \big[g_r^- \leqslant s \leqslant g_r^+\big]$, что, опять-таки, совпадает с (9.4).

Заметим также, что при $s = 0$ запись $G^{s-1}_{m-1} = 0$ по определению корректна, в то же время $H^{s-1}_{m-1} = 0$, поскольку $N_1 = \varnothing$. Аналогично, при $s = m$ имеем $N_0 = \varnothing$, следовательно, $H^s_{m-1} = 0 = G^s_{m-1}$.

3. Односторонние оценки (9.5) и (9.6) выводится аналогично. Различие в том, что для них выражение под знаком произведения в (9.9) принимает вид, соответственно, либо $\bigl[0 \leqslant B_m \leqslant g^+_m(\varepsilon)\bigr]$, либо $\bigl[g^-_m(\varepsilon) \leqslant B_m \leqslant m\bigr]$. Изменяется только форма границы в усечённом треугольнике Паскаля, соответственно, либо нижней $g^-_m(\varepsilon) = 0$, либо верхней $g^+_m(\varepsilon) = m$, и все дальнейшие рассуждения остаются в силе. ■

Геометрическая интерпретация. Вторую часть доказательства (после формулы (9.9)) можно провести гораздо короче и нагляднее, пользуясь следующими геометрическими соображениями.

Каждое разбиение $X \sqcup \bar{X} = \mathbb{X}$ взаимно однозначно соответствует бинарному вектору $b = (b_1, \ldots, b_L)$, состоящему из ℓ единиц и k нулей, и, в то же время, некоторой траектории, проходящей из точки $(0, 0)$ в точку (L, ℓ) согласно правилу: если $b_i = 1$, то сместиться вправо и вверх на 1; если $b_i = 0$, то сместиться вправо на 1, см. рис. 2.3. Очевидно, траектория состоит из всех точек $(m, B_m)^L_{m=0}$. Выполнение совокупности условий $\bigl[g^-_m \leqslant B_m(X) \leqslant g^+_m\bigr]$ при всех $m = 1, \ldots, L$ означает, что траектория не может проходить ниже границы g^-_m или выше границы g^+_m. На рис. 2.3 эти границы показаны линиями. Согласно (9.9) функционал $\mathsf{P}\bigl[D(X) \leqslant \varepsilon\bigr]$ в точности равен доле таких траекторий. Будем называть их *допустимыми*. Обозначим через H^s_m число допустимых траекторий, проходящих из точки $(0, 0)$ в точку (m, s). Допустимая траектория может придти в (m, s) либо из $(m-1, s-1)$, либо из $(m-1, s)$. Отсюда следует рекуррентная формула для числа допустимых траекторий: $H^s_m = H^s_{m-1} + H^{s-1}_{m-1}$. Однако, если $s \notin \bigl[g^-_m, g^+_m\bigr]$, то все такие траектории уже не будут допустимыми, поэтому окончательная формула принимает вид $H^s_m = \bigl(H^s_{m-1} + H^{s-1}_{m-1}\bigr)\bigl[g^-_m \leqslant s \leqslant g^+_m\bigr]$, что совпадает с определением усечённого треугольника Паскаля: $H^s_m \equiv G^s_m$.

Практическое вычисление по рекуррентным соотношениям (9.4) сталкивается с проблемой переполнения: значения G^s_m выходят за пределы разрядной сетки современных компьютеров при L порядка нескольких сотен. Проблема снимается, если вывести рекуррентную формулу для отношений $\varphi^s_m = G^s_m / C^s_m$, которые принимают значения из отрезка $[0, 1]$. Применив тождества $C^s_m = \frac{m}{m-s} C^{s-1}_m = \frac{m}{s} C^{s-1}_{m-1}$, получим:

$$\varphi^s_m = \frac{m-s}{m} \varphi^s_{m-1} + \frac{s}{m} \varphi^{s-1}_{m-1}.$$

Усечённый треугольник Паскаля оказывается полезной концепцией не только при выводе точного выражения для критерия Смирнова, но во многих задачах, связанных со случайными блужданиями при ограничениях. Упомянем только выборочный контроль качества [3] и анализ выживаемости [32].

§9.4 Обобщение на случай вариационного ряда со связками

В Теореме 9.1 (Смирнова) требование непрерывности функции распределения является существенным. В сильной аксиоматике оно гарантирует, что с вероятно-

стью 1 вариационный ряд $\xi(x^{(1)}) < \xi(x^{(2)}) < \cdots < \xi(x^{(L)})$ не содержит одинаковых элементов, следовательно, ранжировка определена единственным образом. Если условие непрерывности нарушается, равенства (9.2) и (9.3) могут не выполняться [4].

В Теореме 9.2 требование различности всех элементов вариационного ряда формулировалось в явном виде. Покажем, оставаясь в рамках слабой аксиоматики, что отказ от этого требования не сильно меняет вид результата — в Теореме 9.2 изменяются только границы усечения $[g_m^-, g_m^+]$. Следующая теорема обобщает критерий Смирнова на случай дискретных распределений.

Теорема 9.3. *Пусть $\xi\colon \mathbb{X} \to \mathbb{R}$ — произвольная функция, $\mathbb{X}$ — произвольная конечная выборка, вариационный ряд значений $\xi(x_i)$ состоит из H связок:*

$$\underbrace{\xi(x^{(1)}) = \cdots = \xi(x^{(i_1)})}_{\text{1-я связка}} < \underbrace{\xi(x^{(i_1+1)}) = \cdots = \xi(x^{(i_2)})}_{\text{2-я связка}} < \cdots < \underbrace{\xi(x^{(i_{H-1}+1)}) = \cdots = \xi(x^{(i_H)})}_{\text{H-я связка}}.$$

Тогда в слабой аксиоматике справедливы точные оценки (9.5), (9.6), (9.7), *если взять границы усечённого треугольника Паскаля $[\tilde g_m^-, \tilde g_m^+]$:*

$$\tilde g_m^+(\varepsilon) = \min\bigl\{g_{i_{h-1}}^+(\varepsilon) + m - i_{h-1},\ g_{i_h}^+(\varepsilon)\bigr\};$$
$$\tilde g_m^-(\varepsilon) = \max\bigl\{g_{i_{h-1}}^-(\varepsilon),\ g_{i_h}^-(\varepsilon) + m - i_h\bigr\};$$

для всех $m = i_{h-1}{+}1, \ldots, i_h$, где h пробегает значения от 1 до H, $i_0 = 0$, $i_H = L$.

Доказательство в целом аналогично доказательству Теоремы 9.2, поэтому остановимся только на различиях.

Доказательство. Рассмотрим выражение (9.8). Как и прежде, изменим порядок слагаемых, просуммировав их в порядке возрастания значений $\xi(x_i)$:

$$D(X) = \max_{z\in\mathbb{R}} \left| \frac{1}{k}\sum_{i=1}^{L}(1-b_i)\bigl[\xi(x^{(i)}) < z\bigr] - \frac{1}{\ell}\sum_{i=1}^{L} b_i\bigl[\xi(x^{(i)}) < z\bigr] \right|.$$

Максимум достаточно брать не по всем $z \in \mathbb{R}$, а лишь по конечному множеству значений, которые функция ξ принимает на выборке, $z \in \bigl\{\xi(x^{(1)}), \ldots, \xi(x^{(H)})\bigr\}$. Уберём сомножитель $\bigl[\xi(x^{(i)}) < z\bigr]$, заменив верхний предел суммирования L на $m = \max\bigl\{i\colon \xi(x^{(i)}) < z\bigr\}$. Заметим, что все объекты одной связки либо вместе входят, либо вместе не входят в сумму по i. Поэтому число m может принимать значения только из множества $I_H = \{i_1, \ldots, i_H\}$:

$$D(X) \;=\; \max_{m\in I_H} \left| \frac{1}{k}\sum_{i=1}^{m}(1-b_i) - \frac{1}{\ell}\sum_{i=1}^{m} b_i \right| \;=\; \frac{L}{\ell k} \max_{m\in I_H} \left| B_m - \frac{m\ell}{L} \right|.$$

Аналогично (9.9), получаем:

$$\mathsf{P}\bigl[D(X) \leqslant \varepsilon\bigr] = \mathsf{P}\prod_{m\in I_H} \bigl[g_m^-(\varepsilon) \leqslant B_m \leqslant g_m^+(\varepsilon)\bigr].$$

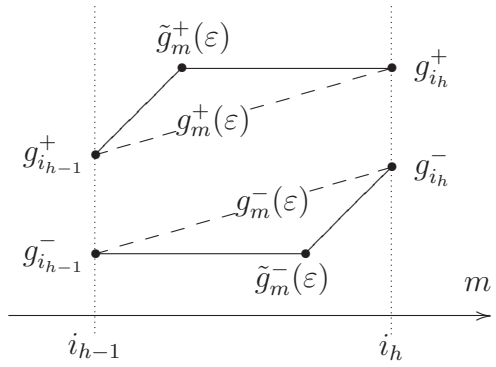


Рис. 9.5. Верхние $\tilde g^+_m(\varepsilon)$ и нижние $\tilde g^+_m(\varepsilon)$ границы усечённого треугольника Паскаля в сравнении с линейными границами $g^-_m(\varepsilon)$ и $g^+_m(\varepsilon)$ на отрезке $m = \{i_{h-1},\dots,i_h\}$, соответствующем h-й связке.

Единственное отличие от (9.9) заключается в том, что прохождение допустимых траекторий $\big(m, B_m\big)_{m=0}^L$ ограничено сверху $g^+_m(\varepsilon)$ и снизу $g^-_m(\varepsilon)$ не во всех точках $m = 0,\dots,L$, а только в точках $m \in I_H$, соответствующих концам связок.

Рассмотрим допустимые траектории на отрезке $m = \{i_{h-1},\dots,i_h\}$, соответствующем h-й связке, см. рис. 9.5.

Между точками верхней границы $\big(i_{h-1}, \lfloor g^+_{i_{h-1}}\rfloor\big)$ и $\big(i_h, \lfloor g^+_{i_h}\rfloor\big)$ допустимая траектория может идти произвольным образом, следовательно, её путь ограничен сверху горизонтальной прямой $B_m \leqslant g^+_{i_h}$ и наклонной прямой $B_m \leqslant g^+_{i_{h-1}} + (m - i_{h-1})$.

Между точками нижней границы $\big(i_{h-1}, \lceil g^-_{i_{h-1}}\rceil\big)$ и $\big(i_h, \lceil g^-_{i_h}\rceil\big)$ допустимая траектория может идти произвольным образом, следовательно, её путь ограничен снизу горизонтальной прямой $B_m \geqslant g^-_{i_{h-1}}$ и наклонной прямой $B_m \geqslant g^-_{i_h} + (m - i_h)$.

Таким образом, получены границы $[\tilde g^-_m, \tilde g^+_m]$ усечённого треугольника Паскаля.

Теорема доказана. ■

Замечание 9.1. Если все связки одноэлементные, $\{i_1,\dots,i_H\} \equiv \{1,\dots,L\}$, то $\tilde g^+_m(\varepsilon) = g^+_m(\varepsilon)$, $\tilde g^-_m(\varepsilon) = g^-_m(\varepsilon)$, и Теорема 9.3 переходит в Теорему 9.2.

Замечание 9.2. Полученные оценки являются точными, но ненаблюдаемыми. Модифицированные границы $[\tilde g^-_m, \tilde g^+_m]$ существенно зависят от последовательности $i_1,\dots,i_H$, которая строится по всей генеральной выборке $\mathbb{X}$; её невозможно знать, имея лишь наблюдаемую выборку X. Это означает, что Теорему 9.3 можно применять для проверки гипотезы однородности, однако непосредственно она не годится для предсказания эмпирической функции распределения.

Резюме

Вероятность большого отклонения эмпирических распределений имеет асимптотическое выражение (9.2) или (9.3), которое приводится во многих учебниках и справочниках по статистике. В слабой аксиоматике она имеет точное выражение через усечённый треугольник Паскаля и легко обобщается на случай вариационного ряда со связками, то есть на распределения с разрывами и дискретные распределения.

В следующей лекции мы вернёмся к проблеме переобучения и рассмотрим функционал равномерного отклонения частоты ошибок в двух выборках. Он является завышенной оценкой вероятности переобучения. Завышенность приводит к тому, что связность семейства для данного функционала учесть можно, а расслоение — нет. Для монотонной цепи алгоритмов будет получена точная оценка функционала равномерного отклонения. Интересен тот факт, что она также связана со случайными блужданиями, и также выражается через усечённый треугольник Паскаля, а в некоторых случаях даже совпадает с оценкой из теоремы Смирнова.

10 Оценки вероятности равномерного отклонения

Функционал вероятности большого *равномерного отклонения* частот ошибок в двух выборках вводится в VC-теории и берётся за основу во многих последующих исследованиях (см. обзоры [58, 34, 10]). Он является верхней оценкой вероятности переобучения и не зависит от метода обучения μ, что даёт ему определённое практическое преимущество. С помощью обращения из него получается верхняя оценка частоты ошибок на контрольной выборке. Её минимизация, в свою очередь, приводит к новому методу обучения μ, который, конечно же, отличается от обычной минимизации эмпирического риска. По идее, новый метод μ должен быть менее подвержен переобучению, поскольку он строился из соображений оптимизации обобщающей способности. С другой стороны, завышенность использованной оценки может повлечь за собой неоптимальность или даже неадекватность нового метода.

Получив несколькими способами верхние оценки вероятности большого равномерного отклонения, мы выясним, что все они учитывают связность, но не учитывают расслоение. Именно в этом проявляется завышенность оценок вероятности переобучения через вероятность равномерного отклонения.

§10.1 Техника порождающих и запрещающих множеств

Напомним, что *переобученностью* алгоритма a на выборке X мы называем разность частоты его ошибок на двух выборках, контрольной и обучающей:

$$\delta(a, X) = \nu(a, \bar{X}) - \nu(a, X).$$

Функционал вероятности переобучения

$$Q_\varepsilon(\mu, \mathbb{X}) = \mathsf{P}\big[\delta(\mu X, X) \geqslant \varepsilon\big]$$

оценивается сверху вероятностью большого равномерного (по множеству алгоритмов A) отклонения частот в двух подвыборках:

$$Q_\varepsilon(\mu, \mathbb{X}) \leqslant \widetilde{Q}_\varepsilon(A, \mathbb{X}) = \mathsf{P} \max_{a \in A} \big[\delta(a, X) \geqslant \varepsilon\big] = \mathsf{P}\Big[\max_{a \in A} \delta(a, X) \geqslant \varepsilon\Big].$$

Метод максимизации переобученности. Принцип порождающих и запрещающих множеств может быть применён к функционалу $\widetilde{Q}_\varepsilon(A, \mathbb{X})$, если специальным образом ввести метод обучения. Эта идея принадлежит И. Толстихину.

Определение 10.1. *Метод обучения μ называется максимизацией переобученности (discrepancy maximization), если* $\mu X = \arg\max_{a \in A} \delta(a, X)$.

Из определения следует, что если метод μ максимизирует переобученность, то вероятность переобучения совпадает с вероятностью равномерного отклонения,

$$Q_\varepsilon(\mu, \mathbb{X}) = \widetilde{Q}_\varepsilon(A, \mathbb{X}).$$

Лемма 10.1. *Если метод μ максимизирует переобученность, то для каждого алгоритма $a \in A$ множество $X_a = \{x_{ab} \in \mathbb{X}\colon a \prec b\}$ является порождающим, а множество $X'_a = \{x_{ba} \in \mathbb{X}\colon b \prec a\}$ — запрещающим в смысле Гипотезы 7.1:*

$$[\mu X = a] \leqslant [X_a \subseteq X]\,[X'_a \subseteq \bar{X}].$$

Доказательство. Из того, что $\mu X = a$ следует

$$\delta(a, X) \geqslant \delta(b, X) \quad \text{для всех } b \in A. \tag{10.1}$$

Возьмём произвольный $x_{ab} \in X_a$. Тогда $I(a, x_{ab}) = 0$, $I(b, x_{ab}) = 1$. Если допустить, что $x_{ab} \in \bar{X}$, то получим

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) + \tfrac{1}{k} - \nu(a, X) > \delta(a, X),$$

что противоречит (10.1), значит, $x_{ab} \in X$. В силу его произвольности $X_a \subseteq X$.

Возьмём произвольный $x_{ba} \in X'_a$. Тогда $I(b, x_{ba}) = 0$, $I(a, x_{ba}) = 1$. Если допустить, что $x_{ba} \in X$, то получим

$$\delta(b, X) = \nu(b, \bar{X}) - \nu(b, X) = \nu(a, \bar{X}) - \nu(a, X) + \tfrac{1}{\ell} > \delta(a, X),$$

что противоречит (10.1), значит, $x_{ba} \in \bar{X}$. В силу его произвольности $X'_a \subseteq \bar{X}$. ∎

Оценка связности. Из Леммы 10.1 и общей оценки расслоения–связности (7.2) следует оценка связности для вероятности равномерного отклонения.

Теорема 10.2. *Пусть векторы ошибок всех алгоритмов из A попарно различны. Тогда имеет место верхняя оценка вероятности равномерного отклонения:*

$$\widetilde{Q}_\varepsilon(A, \mathbb{X}) \leqslant \sum_{a \in A} \frac{C^{\ell-q}_{L-q-d}}{C^\ell_L} H^{\ell-q,\ m-d}_{L-q-d}\left(\tfrac{\ell}{L}(m - \varepsilon k)\right), \tag{10.2}$$

где $q = q(a) = |X_a|$ — верхняя связность, $d = d(a) = |X'_a|$ — нижняя связность алгоритма a, $m = n(a, \mathbb{X})$ — число ошибок алгоритма a на генеральной выборке.

Эта оценка гораздо хуже аналогичной по структуре оценки расслоения–связности (7.5) для пессимистичной МЭР. Единственное различие заключается в том, что неоптимальность $r(a)$ заменяется на нижнюю связность $d(a)$. Неоптимальность растёт с номером слоя линейно, соответственно, вклады алгоритмов в Q_ε убывают экспоненциально. Нижняя связность $d(a)$ не превышает $r(a)$ и, как показывают эксперименты [23], концентрируется вокруг одного и того же значения во всех слоях, поэтому алгоритмы всех слоёв вносят примерно равный вклад в оценку (10.2). Таким образом, эта оценка учитывает связность, но не учитывает расслоение.

§10.2 Техника цепных разложений

Вероятность равномерного отклонения можно оценивать сверху с помощью *цепных разложений* (chain expansion). Эта техника предложена Д. Кочедыковым [23].

Теорема 10.3. *Для любых* $\mathbb{X}$, A *и любого* $\varepsilon \in [0,1]$

$$\widetilde{Q}_\varepsilon(A,\mathbb{X}) \leqslant \sum_{a\in A} \Big[\big\{s_a(\varepsilon)\big\} < \tfrac{\ell}{L}\Big] \frac{C^\ell_{L-d}}{C^\ell_L} h^{\ell,\,m-d}_{L-d}\big(s_a(\varepsilon)\big)\,. \tag{10.3}$$

где $d = d(a)$ *— нижняя связность алгоритма* a, $m = n(a,\mathbb{X})$ *— число ошибок алгоритма* a *на генеральной выборке,* $s_a(\varepsilon) = \frac{\ell}{L}\big(m - \varepsilon k\big)$,

Доказательство. Обозначим через $U_a(X,\varepsilon)$ или просто U_a индикатор переобученности алгоритма a на выборке X:

$$U_a = \big[\delta(a,X) \geqslant \varepsilon\big] = \big[n(a,X) \leqslant s_a(\varepsilon)\big].$$

Обозначим через $\bar{U}_a$ отрицание U_a, то есть $\bar{U}_a = 1 - U_a$.

Воспользуемся тем, что на множестве алгоритмов A задано отношение частичного порядка как естественное отношение порядка на булевых векторах ошибок алгоритмов. Дополним это отношение до линейного порядка произвольным образом. Представим вероятность максимума бинарных величин как сумму вероятностей, воспользовавшись *цепным разложением*:

$$\widetilde{Q}_\varepsilon = \mathsf{P} \max_a U_a = \sum_{a\in A} \mathsf{P}\, U_a \prod_{\substack{b\in A\\ b\prec a}} \bar{U}_b.$$

Оставим в цепном разложении только такие сомножители $\bar{U}_b$, что $b \prec a$. Остальные сомножители тривиально оценим сверху единицей, $\bar{U}_b \leqslant 1$, $b \not\prec a$. Тогда

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} \mathsf{P} \prod_{\substack{b\in A\\ b\prec a}} U_a \bar{U}_b.$$

Рассмотрим произведение $U_a\bar{U}_b$ для произвольных b и a таких, что $b \prec a$:

$$\begin{aligned}
U_a\bar{U}_b &= \big[n(a,X) \leqslant s_a(\varepsilon)\big]\big[n(b,X) > s_b(\varepsilon)\big] = \\
&= \big[s_a(\varepsilon) - \tfrac{\ell}{L} = s_b(\varepsilon) < n(b,X) \leqslant n(a,X) \leqslant s_a(\varepsilon)\big] = \\
&= \big[s_a(\varepsilon) - \tfrac{\ell}{L} < n(a,X) \leqslant s_a(\varepsilon)\big]\big[n(b,X) = n(a,X)\big] = \\
&= \big[n(a,X) = \lfloor s_a(\varepsilon)\rfloor\big]\big[\{s_a(\varepsilon)\} < \tfrac{\ell}{L}\big]\big[x_{ba} \in \bar{X}\big],
\end{aligned}$$

где $\lfloor s\rfloor$ — целая часть s, $\{s\}$ — дробная часть s. Тогда

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} \mathsf{P} \prod_{\substack{b\in A\\ b\prec a}} U_a\bar{U}_b = \sum_{a\in A} \mathsf{P} \prod_{\substack{b\in A\\ b\prec a}} \big[n(a,X) = \lfloor s_a(\varepsilon)\rfloor\big]\big[\{s_a(\varepsilon)\} < \tfrac{\ell}{L}\big]\big[x_{ba} \in \bar{X}\big].$$

Сомножители, не зависящие от b, вынесем за знак произведения. Введём множество $X'_a = \{x_{ba} \in \mathbb{X}\colon b \prec a\}$ и заметим, что

$$\prod_{\substack{b\in A\\ b\prec a}} [x_{ba} \in \bar{X}] = [X'_a \subseteq \bar{X}].$$

Таким образом,

$$\widetilde{Q}_\varepsilon \leqslant \sum_{a\in A} [\{s_a(\varepsilon)\} < \tfrac{\ell}{L}]\, \mathsf{P}[n(a,X) = \lfloor s_a(\varepsilon)\rfloor]\, [X'_a \subset \bar{X}].$$

Учитывая, что $|X'_a| = d(a)$, получим оценку (10.2). Теорема доказана. ■

Оценка (10.3) по своей структуре аналогична оценке (10.2). В ней есть два улучшения: во-первых, суммирование производится не по всем слоям семейства алгоритмов (при $\ell = k$ учитывается только каждый второй слой), во-вторых, вместо «левого хвоста» гипергеометрического распределения H в ней фигурирует значение h гипергеометрического распределения в одной точке. С другой стороны, оценка (10.3) учитывает только нижнюю связность, что в итоге делает её более слабой по сравнению с оценкой (10.2). При этом обе оценки не учитывают расслоение.

§10.3 Техника случайных блужданий

Для произвольного семейства A выразим равномерное отклонение частоты ошибок в выборках X и $\bar{X}$ через индикаторы $b_i = b_i(X) = [x_i \in X]$:

$$\begin{aligned} D(X) &= \max_{a\in A} \delta(a,X) = \max_{a\in A} \big(\nu(a,\bar{X}) - \nu(a,X)\big) = \\ &= \max_{a\in A} \sum_{i=1}^{L} I(a,x_i) \Big(\tfrac{1}{k} \underbrace{[x_i \in \bar{X}]}_{1-b_i} - \tfrac{1}{\ell} \underbrace{[x_i \in X]}_{b_i} \Big) = \\ &= \frac{L}{\ell k} \max_{a\in A} \sum_{i=1}^{L} I(a,x_i) \left(\frac{\ell}{L} - b_i \right). \end{aligned} \tag{10.4}$$

Монотонная цепь алгоритмов — это модельное семейство $A = \{a_0, a_1, \ldots, a_D\}$, в котором алгоритм a_0 допускает m ошибок на генеральной выборке, а каждый последующий алгоритм хуже предыдущего на одном объекте (определение 6.7, стр. 52). Перенумеруем объекты $x_1, \ldots, x_L$ таким образом, чтобы

$$I(a_d, x_i) = [i \leqslant m + d], \quad i = 1, \ldots, L, \quad d = 0, \ldots, D. \tag{10.5}$$

Теорема 10.4. *Для монотонной цепи алгоритмов A, произвольной генеральной выборки $\mathbb{X}$ и произвольного $\varepsilon \in [0,1]$ справедлива точная оценка вероятности того, что равномерное отклонение не превысит ε:*

$$\mathsf{P}\Big[\max_{a\in A} \delta(a,X) \leqslant \varepsilon\Big] = \frac{G_L^\ell\big[g_L^-(\varepsilon), L\big]}{C_L^\ell},$$

где $g_t^-(\varepsilon)$ — левая граница усечённого треугольника Паскаля:

$$g_t^-(\varepsilon) = \tfrac{\ell}{L}(t - \varepsilon k)\big[m \leqslant t \leqslant m + D\big].$$

Доказательство. В случае монотонной цепи представление (10.4) упрощается после подстановки в него (10.5):

$$D(X) = \frac{L}{\ell k} \max_{d=0..D} \sum_{i=1}^{m+d} \left(\frac{\ell}{L} - b_i \right) = \frac{L}{\ell k} \max_{d=0..D} \left(\frac{\ell(m+d)}{L} - B_{m+d} \right),$$

где $B_{m+d} = b_1 + \cdots + b_{m+d}$. Таким образом, равномерное отклонение частоты ошибок в выборках $\bar{X}$ и X выражается через равномерное отклонение числа единиц среди первых $m+d$ членов бинарной последовательности $b_1, \ldots, b_L$ от «ожидаемого» числа единиц $\frac{\ell}{L}(m+d)$.

Найдём теперь вероятность того, что $D(X)$ не превышает ε:

$$\begin{aligned}
\mathsf{P}\big[D(X) \leqslant \varepsilon\big] &= \mathsf{P}\Big[\max_{d=0..D} \big(\tfrac{\ell}{L}(m+d) - B_{m+d}\big) \leqslant \tfrac{\ell}{L}\varepsilon k\Big] = \\
&= \mathsf{P} \prod_{d=0}^{D} \Big[\tfrac{\ell}{L}\big(m+d-\varepsilon k\big) \leqslant B_{m+d}\Big] = \\
&= \mathsf{P} \prod_{t=1}^{L} \big[g_t^-(\varepsilon) \leqslant B_t\big].
\end{aligned}$$

Полученное выражение в точности совпадает с вероятностью равномерного отклонения эмпирических распределений (9.9), для которой уже найдено точное выражение (9.6) через усечённый слева треугольник Паскаля. Теорема доказана. ■

Резюме

Функционал равномерного отклонения, введённый в теории Вапника-Червоненкиса, учитывает связность семейства алгоритмов, но не учитывает его расслоение. Из-за этого он может давать сильно завышенные оценки вероятности переобучения. Тем не менее, он активно используется в теории статистического обучения. Его достоинство в том, что он не зависит от метода обучения, что позволяет разрабатывать новые методы обучения путём минимизации его обращённых верхних оценок.

При получении верхних оценок вероятности равномерного отклонения могут накапливаться дополнительные погрешности. Среди трёх рассмотренных техник первые две дают завышенные оценки, третья даёт точную оценку, но только для частного случая монотонной цепи алгоритмов.

В следующей лекции мы вернёмся к модельным семействам алгоритмов и рассмотрим несколько нетривиальных частных случаев, когда оценки расслоения–связности оказываются точными.

Упражнения

Задача 10.1 (10*). Получить точную оценку вероятности равномерного отклонения $\widetilde{Q}_\varepsilon(A, \mathbb{X})$.

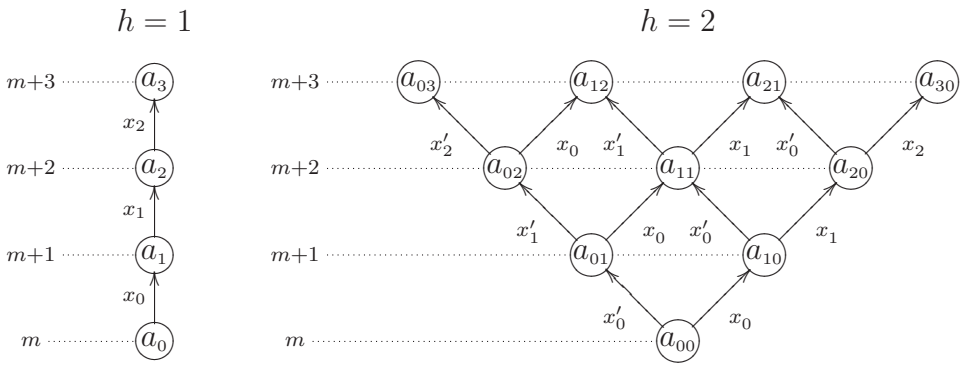
11 Точные оценки вероятности переобучения

Существуют примеры нетривиальных модельных семейств алгоритмов, для которых оценки расслоения–связности являются точными. Среди них многомерные монотонные сети алгоритмов и интервалы булева куба, обладающие теми же ключевыми свойствами расслоения и связности, что и реальные семейства.

§11.1 Многомерная монотонная сеть алгоритмов

В §6.5 мы рассмотрели монотонную цепь алгоритмов, которая является моделью однопараметрического семейства алгоритмов. Естественным обобщением этой модели на многомерный случай представляется семейство, граф расслоения–связности которого изоморфен прямоугольной решётке заданной размерности h или некоторому её невырожденному фрагменту той же размерности.

Двумерный случай. На рисунке показаны четыре нижних слоя графов расслоения–связности размерностей 1 (слева) и 2 (справа). Горизонтальные линии соответствуют слоям, m — число ошибок лучшего алгоритма на генеральной выборке. Объекты вдоль стрелок означают, что при переходе от нижнего алгоритма к верхнему ошибка появляется именно на данном объекте.



Рассмотрим произвольный алгоритм a_{uv} из слоя $m + t = n(a_{uv}, \mathbb{X})$. Число алгоритмов в этом слое равно $t + 1$. Верхняя связность $q(a_{uv}) = 2$ совпадает с размерностью $h = 2$. Неоптимальность $r(a_{uv})$ равна t, поскольку a_{uv} отличается от лучшего алгоритма a_{00} ровно на t объектах. Подставив все эти данные в оценку (7.5) и заменив сумму по алгоритмам суммой по слоям, получим оценку расслоения–связности для T нижних слоёв двумерной сети алгоритмов:

$$Q_\varepsilon \leqslant \sum_{t=0}^{T} (t+1) \frac{C_{L-t-2}^{\ell-2}}{C_L^\ell} H_{L-t-2}^{\ell-2,\,m}\left(\tfrac{\ell}{L}(m+t-\varepsilon k)\right).$$

Пока это были предварительные соображения, показывающие, что иногда оценки расслоения–связности получаются очень просто. Эта оценка не вполне корректна, так как при подсчёте верхней связности алгоритмов T-го слоя предполагалось, что $(T+1)$-й слой существует, но его представители в сумме не учтены. Далее мы

введём необходимый формализм, обобщим оценку на случай произвольной размерности и аккуратно разберёмся с верхними слоями. Мы также убедимся, что оценка расслоения–связности для h-мерной монотонной сети является точной. Первым эту оценку получил П. Ботов [5].

Случай произвольной размерности. Пусть $J = (j_1, \ldots, j_h)$ — целочисленный вектор индексов. Положим $|J| = j_1 + \cdots + j_h$. Введём на векторах индексов отношение частичного порядка: для произвольных $J = (j_1, \ldots, j_h)$ и $K = (k_1, \ldots, k_h)$ положим $J \leqslant K$, если $j_d \leqslant k_d$ для всех $d = 1, \ldots, h$. Положим $J < K$, если $J \leqslant K$ и $J \neq K$.

Определение 11.1. *Монотонной сетью алгоритмов размерности h и высоты H называется множество алгоритмов $A = \{a_J \colon |J| \leqslant H\}$ с попарно различными векторами ошибок, такое, что*

1) если $J < K$, то $a_J < a_K$;

2) $n(a_J, \mathbb{X}) = m + |J|$ при некотором m.

Алгоритм $a_0 = a_{(0,\ldots,0)}$ называется лучшим в A.

Данное определение может показаться неконструктивным, так как оно не описывает матрицу ошибок в явном виде. Однако следующая лемма показывает, что на самом деле матрица ошибок определена однозначно.

Лемма 11.1. *Множество объектов $\mathbb{X}$ распадается на объекты трёх типов:*

1) m объектов, на которых ошибаются все алгоритмы;

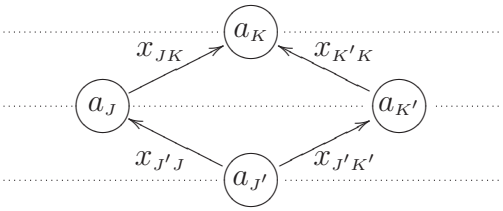
2) Hh объектов $\{x_t^d \colon t = 0, \ldots, H-1,\ d = 1, \ldots, h\}$ таких, что $I(a_J, x_t^d) = [t < j_d]$;

3) все остальные объекты, на которых ни один из алгоритмов не ошибается.

Доказательство. 1. Множество объектов, на которых ошибаются все алгоритмы, совпадает с множеством объектов, на которых ошибается лучший алгоритм a_0, и их число равно $m = n(a_0, \mathbb{X})$. Что и утверждается в первом пункте леммы.

Введём на векторах индексов покомпонентное сложение и умножение на число: $J + K = (j_1 + k_1, \ldots, j_h + k_h)$ и $cJ = (cj_1, \ldots, cj_h)$ для произвольных $J = (j_1, \ldots, j_h)$ и $K = (k_1, \ldots, k_h)$. Определим единичные векторы $E_d = \big([d{=}s]\big)_{s=1}^h$, $d = 1, \ldots, h$.

2. Возьмём произвольный алгоритм $a_J \in A$ и две размерности $\{d, s\} \subseteq \{1, \ldots, h\}$ такие, что $j_d < H$, $j_s > 0$. Построим векторы индексов $K = J + E_d$, $J' = J - E_s$, $K' = K - E_s$. Четыре алгоритма a_J, a_K, $a_{J'}$, $a_{K'}$ образуют граф расслоения–связности (напомним, что если алгоритм a предшествует алгоритму b, $a \prec b$, то через x_{ab} обозначается тот единственный объект, для которого $I(a, x_{ab}) = 0$ и $I(b, x_{ab}) = 1$):



Алгоритмы $a_{J'}$ и a_K отличаются только на двух объектах, поэтому среди четырёх объектов $x_{J'J}$, $x_{J'K'}$, x_{JK}, $x_{K'K}$ имеются только два различных. Единственный допустимый вариант совпадений — когда $x_{JK} = x_{J'K'}$ и $x_{J'J} = x_{K'K}$. Продолжая

по очереди уменьшать в векторах J и K все координаты $j_1, \ldots, j_h$ кроме j_d, придём в итоге к тому, что два алгоритма $a_{j_d E_d} \prec a_{(j_d+1)E_d}$ также отличаются на объекте x_{JK}. Таким образом, объект x_{JK} определяется только координатой $d \in \{1, \ldots, h\}$ и значением индекса $t = j_d \in \{0, \ldots, H-1\}$, и не зависит от остальных координат вектора J. Обозначим этот объект через x_t^d.

Докажем от противного, что среди объектов x_t^d нет совпадающих. Пусть $x_t^d = x_{t'}^d$ и для определённости $t < t'$. Тогда $a_{tE_d} < a_{t'E_d}$, следовательно $I(a_{t'E_d}, x_t^d) = 1$. С другой стороны, $I(a_{t'E_d}, x_{t'}^d) = 0$. Поэтому объекты x_t^d и $x_{t'}^d$ не могут совпадать. Предположим теперь, что $x_t^d = x_{t'}^{d'}$ при $d \neq d'$. Возьмём алгоритм a_J, $J = tE_d + t'E_{d'}$. Алгоритмы a_{J+E_d} и $a_{J+E_{d'}}$ отличаются от a_J на объектах x_t^d и $x_{t'}^{d'}$ соответственно и должны иметь различные векторы ошибок по определению монотонной сети. Поэтому объекты x_t^d и $x_{t'}^{d'}$, опять-таки, не могут совпадать.

Итак, произвольный алгоритм $a_J \in A$ допускает ошибки на тех и только тех объектах x_t^d, у которых $t < j_d$, что и утверждается во втором пункте леммы.

3. В выборке могут оставаться объекты x, на которых алгоритм a_0 не допускает ошибку, и которые не совпадают ни с одним из x_t^d. Если бы один из алгоритмов $a \in A$ допускал ошибку на таком объекте x, то существовали бы алгоритмы $b, c \in A$, такие, что $b \prec c \leqslant a$ и $x = x_{bc}$. Но тогда объект x был бы одним из объектов x_t^d, как было показано выше. Таким образом, ни один из алгоритмов $a \in A$ не может допускать ошибку на таком объекте x. Что и утверждается во третьем пункте леммы.

Лемма доказана. ■

Следствие 11.1.1. *Для любого $a_J \in A$, $J = (j_1, \ldots, j_d, \ldots, j_h)$ и любого $d \in \{1, \ldots, h\}$ увеличение индекса $t = j_d$ на единицу, $K = (j_1, \ldots, j_d{+}1, \ldots, j_h)$ приводит к появлению ошибки на объекте x_t^d:*

$$a_J \prec a_K; \qquad I(a_J, x_t^d) = 0, \quad I(a_K, x_t^d) = 1.$$

Интерпретация. Монотонная сеть размерности h и высоты H — это модель семейства алгоритмов с h непрерывными параметрами. Модельное предположение заключается в том, что по мере увеличения d-го параметра ошибки возникают последовательно на объектах $x_0^d, \ldots, x_{H-1}^d$, независимо от значений остальных параметров. В реальных задачах это требование как правило, не выполняется. В то же время, известны примеры, когда монотонная сеть порождается искусственной выборкой и реальным семейством алгоритмов (упражнения 11.1, 11.2).

Размерные характеристики. Из Леммы 11.1 следует, что необходимым условием существования монотонной сети является ограничение $m + Hh \leqslant L$.

Число алгоритмов в слое $m + t$ равно числу неотрицательных целочисленных векторов индексов $J = (j_1, \ldots, j_h)$ таких, что $|J| = j_1 + \ldots + j_h = t$. Оно же есть число способов выбрать t предметов из h с повторениями, $\bar{C}_h^t = C_{h+t-1}^{h-1}$. Просуммировав число алгоритмов по слоям и воспользовавшись известным комбинаторным тождеством, нетрудно найти число алгоритмов во всей сети:

$$|A| = C_{h-1}^{h-1} + \cdots + C_{h+H-1}^{h-1} = C_{H+h}^{h}.$$

Оценка расслоения–связности. Из формул (7.3), (7.4) и графа расслоения–связности следует, что для любого алгоритма a_J из монотонной сети A

$X_J = \{x_t^d \colon t = j_d,\ d = 1, \ldots, h\}$ — порождающее множество;
$X'_J = \{x_t^d \colon t < j_d,\ d = 1, \ldots, h\}$ — запрещающее множество.

Лемма 7.1 даёт лишь необходимое условие того, что алгоритм a_J будет получен как результат обучения: $[\mu X{=}a_J] \leqslant [X_J \subseteq X]\,[X'_J \subseteq \bar{X}]$. Однако в случае многомерной монотонной сети необходимое условие является также и достаточным.

Лемма 11.2. *Если μ — пессимистичная минимизация эмпирического риска, то*

$$[\mu X{=}a_J] = [X_J \subseteq X]\,[X'_J \subseteq \bar{X}] \quad \textit{для каждого } a_J \in A.$$

Доказательство. Докажем, что если $X_J \subseteq X$ и $X'_J \subseteq \bar{X}$, то только алгоритм a_J может быть результатом обучения.

Алгоритм a_J допускает ошибки только на объектах из X'_J, но все они лежат в контроле $\bar{X}$, поэтому $n(a_J, X) = 0$.

Рассмотрим произвольный алгоритм a_K такой, что $K \not\leqslant J$. Это означает, что $k_d > j_d$ для некоторой координаты d. Но тогда a_K допускает ошибку на объекте $x_{j_d}^d$, который лежит в X_J, следовательно, и в обучении X. Поэтому алгоритм a_K не может быть выбран методом μ, минимизирующим эмпирический риск.

Рассмотрим произвольный алгоритм a_K такой, что $K < J$. Тогда оба алгоритма, a_J и a_K не допускают ошибок на обучении X. Допустим, что $k_d < j_d$ для некоторой координаты d. Тогда на контрольном объекте $x_{k_d}^d$ из двух алгоритмов только a_J допускает ошибку, поэтому метод μ, будучи пессимистичным, выберет алгоритм a_J.

Таким образом, только алгоритм a_J может быть результатом обучения. ■

Теорема 11.3 (О монотонной сети алгоритмов). *Пусть A — монотонная сеть размерности h и высоты H, метод μ является пессимистичной минимизацией эмпирического риска и выполнено условие $m + Hh \leqslant L$. Тогда*

$$Q_\varepsilon(\mu, \mathbb{X}) = \sum_{t=0}^{\min\{H,k\}} \frac{C_{h+t-1}^t C_{L-q-t}^{\ell-q}}{C_L^\ell} H_{L-q-t}^{\ell-q,\,m}\left(\tfrac{\ell}{L}(m + t - \varepsilon k)\right), \quad q = [t{<}H]h.$$

Доказательство. Возьмём произвольный алгоритм a_J и рассмотрим три случая.

1. Если $|J| > k$, то число ошибок алгоритма a_J на запрещающих объектах X'_J превышает длину контрольной выборки. Часть ошибок обязательно окажется в обучающей подвыборке X, и метод μ выберет другой алгоритм. В этом случае

$$[\mu X{=}a_J] = 0.$$

2. Если $|J| \leqslant k$ и $|J| = H$, то есть алгоритм a_J находится в последнем слое, то порождающих объектов для a_J не существует, следовательно,

$$[\mu X{=}a_J] = [X'_J \subseteq \bar{X}].$$

В этом случае верхняя связность $q(a_J) = 0$, неоптимальность $r(a_J) = |J|$.

3. Если $|J| \leqslant k$ и $|J| < H$, то реализуется основной случай из Леммы 11.2:

$[\mu X{=}a_J] = [X_J \subseteq X][X'_J \subseteq \bar{X}]$.

В этом случае верхняя связность $q(a_J) = |X_J| = h$, неоптимальность $r(a_J) = |J|$.

Применим оценку расслоения–связности (7.5), которая в данном случае является точным равенством:

$$Q_\varepsilon = \sum_{J\colon |J|\leqslant H} P_J H_{L-q-r}^{\ell-q,\,m+|J|-r}\left(\tfrac{\ell}{L}\big(m+|J|-\varepsilon k\big)\right), \quad P_J = \frac{C_{L-q-r}^{\ell-q}}{C_L^\ell},$$

где $P_J = \mathsf{P}[\mu X = a_J]$ — вероятность получить алгоритм a_J в результате обучения, $q = q(a_J)$ — верхняя связность, $r = r(a_J) = |J|$ — неоптимальность алгоритма a_J.

Верхняя связность и неоптимальность в каждом слое одинаковы у всех алгоритмов. Это позволяет выразить вероятность того, что методом μ будет получен некоторый алгоритм a_J из слоя $m + t$, где $t = |J|$:

$$P_t = \mathsf{P}\big[n(\mu X, \mathbb{X}) = m{+}t\big] = C_{h+t-1}^t P_J = \frac{C_{h+t-1}^t C_{L-q-t}^{\ell-q}}{C_L^\ell}, \tag{11.1}$$

и затем заменить сумму по алгоритмам суммой по слоям $t = 0, \ldots, H$. Наконец, остаётся учесть, что при $t > k$ алгоритмы вообще не вносят вклад в оценку, а при $t = H$ обнуляется верхняя связность q, которая в остальных случаях равна h.

Таким образом,

$$Q_\varepsilon = \sum_{t=0}^{\min\{H,k\}} P_t H_{L-q-t}^{\ell-q,\,m}\left(\tfrac{\ell}{L}(m+t-\varepsilon k)\right), \quad q = [t{<}H]h,$$

что и требовалось доказать. ■

Эксперимент 1: вычисление оценки расслоения–связности. На рис. 11.1 слева показан график зависимости вероятности P_t получить алгоритм из слоя $m + t$, вычисленной по формуле (11.1), от числа t. Справа показан график зависимости вероятности переобучения Q_ε от порога ε. На каждом графике изображены 9 кривых, соответствующих различным значениям размерности $h = 1, \ldots, 9$.

Зависимость P_t многомерной сетки на отрезке $[0, H - 1]$ унимодальна. Вероятность последнего слоя немного выше, так как для него верхняя связность равна нулю. При росте размерности h положение максимума P_t смещается вправо и растёт вероятность переобучения. Это означает, что рост связности, который приводит к экспоненциальному уменьшению вероятностей P_J отдельных алгоритмов a_J, всё же компенсируется ростом мощности слоёв.

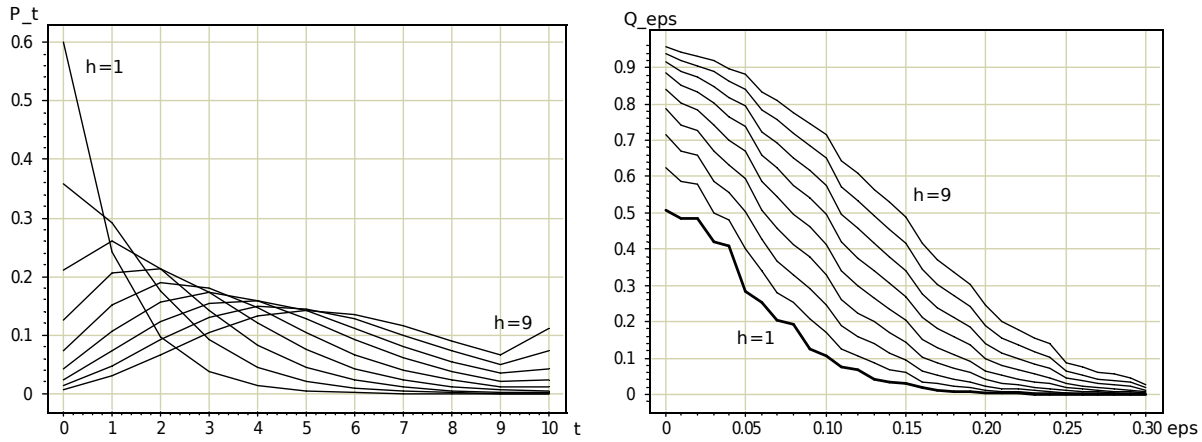


Рис. 11.1. Зависимости P_t от t (слева) и Q_ε от ε (справа) для монотонных сеток размерностей $h = 1, \ldots, 9$ при $L = 100,\ \ell = 60,\ m = 10,\ H = 10$.

Эксперимент 2: сравнение монотонных сеток с реальными семействами. На основе платформы RapidMiner были получены экспериментальные зависимости Q_ε от ε для задач из репозитория UCI (sonar, breast-cancer-wisconsin) на методах классификации NaiveBayes, SVM, DecisionTree, NeuralNetwork [5]. Экспериментальные кривые переобученности $\hat{Q}_\varepsilon$, полученные методом Монте-Карло, аппроксимировались кривыми переобученности монотонных сеток путём подбора размерности h при фиксированных L, ℓ и m. Оказалось, что чем сложнее семейство алгоритмов, тем выше проходит кривая переобученности и тем выше размерность h аппроксимирующей монотонной сетки, см. рис. 11.2, 11.3, 11.4 («более ступенчатые» кривые соответствуют монотонным сеткам, «более гладкие» — реальным семействам).

Примечателен тот факт, что на задаче breast-cancer-wisconsin метод NaiveBayes показал кривую переобученности, эквивалентную монотонной сетке размерности 1, рис. 11.4. Эффективная размерность данной задачи и в самом деле близка к 1, так как очень хорошим разделяющим признаком является сумма всех 9 признаков. DecisionTree на тех же данных не смог найти этот признак и показал бóльшую размерность, бóльшую переобученность и бóльшую ошибку на контроле.

§11.2 Интервал булева куба и его расслоение

Предположим, что векторы ошибок всех алгоритмов из A попарно различны и образуют интервал ранга m в L-мерном булевом кубе. Это означает, что объекты делятся на три группы: m_0 «внутренних» объектов, на которых ни один из алгоритмов не допускает ошибок; m_1 «шумовых» объектов, на которых все алгоритмы допускают ошибки; и m «пограничных» объектов, на которых реализуются все 2^m вариантов допустить ошибки. Других объектов нет: $m_0 + m_1 + m = L$. Алгоритмы допускают от m_1 до $m_1 + m$ ошибок. Число алгоритмов в A равно 2^m. Интервал булева куба обладает свойствами расслоения и связности. На рис. 11.5 показан пример матрицы ошибок для интервала булева куба ранга $m = 7$.

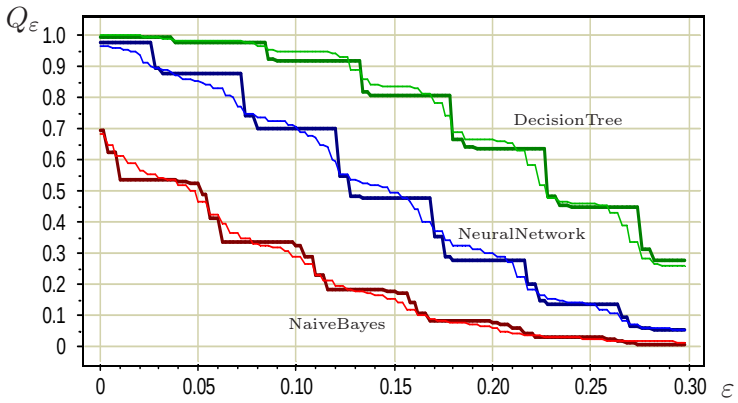


Рис. 11.2. Вероятность переобучения Q_ε алгоритмов NaiveBayes, NeuralNetwork, DecisionTree и монотонных сеток размерностей, соответственно, $h = 8, 28, 46$. Средняя частота ошибок на обучении/контроле, соответственно: 0.28/0.32; 0.05/0.20; 0.02/0.26. Задача sonar, 20 признаков, $L = 208$, $\ell = 187$.

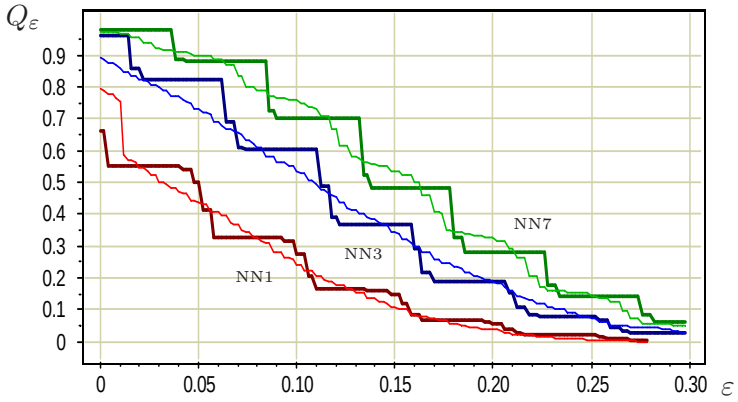


Рис. 11.3. Вероятность переобучения Q_ε алгоритмов NeuralNetwork с числом нейронов в скрытом слое $1, 3, 7$ и монотонных сеток размерностей, соответственно, $h = 8, 22, 30$. Средняя частота ошибок на обучении/контроле, соответственно: 0.29/0.34; 0.12/0.23; 0.03/0.19. Задача sonar, 20 признаков, $L = 208$, $\ell = 187$.

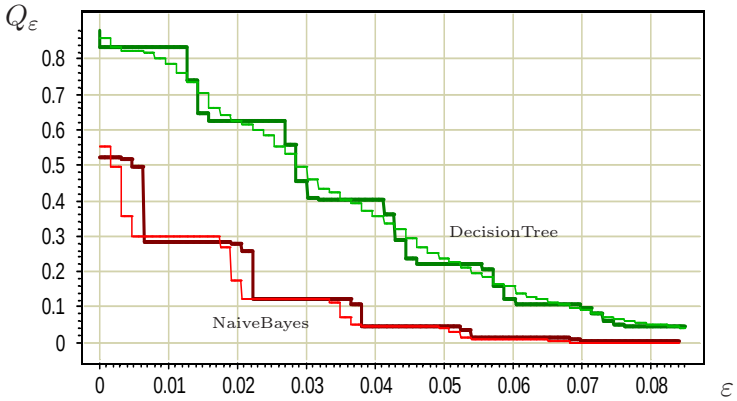


Рис. 11.4. Вероятность переобучения Q_ε алгоритмов NaiveBayes, DecisionTree и монотонных сеток размерности $h = 1, 20$. Средняя частота ошибок на обучении/контроле, соответственно: 0.04/0.04; 0.03/0.06. Задача breast-cancer-wisconsin, 9 признаков, $L = 700$, $\ell = 630$.

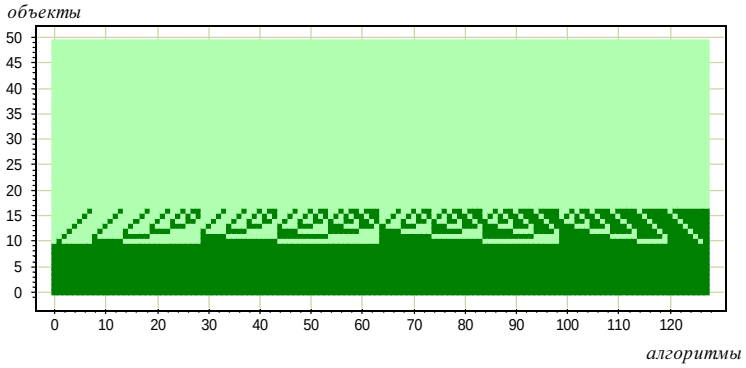


Рис. 11.5. Матрица ошибок интервала булева куба при $L = 50,\ m = 7,\ m_1 = 10$. Число алгоритмов в семействе $|A_m| = 2^m = 128$. Алгоритмы пронумерованы по слоям слева направо, например, A_2 — это первые 29 алгоритмов, A_3 — первые 64.

Для большей общности рассмотрим множество алгоритмов A_t, образованное нижними t слоями интервала булева куба. Это алгоритмы, допускающие не более t ошибок на пограничных объектах, $|A_t| = C_m^0 + C_m^1 + \cdots + C_m^t$. Параметр t может принимать значения $0, \ldots, m$. При $t = 0$ имеем единственный алгоритм, при $t = m$ — полный интервал ранга m. Это модельное семейство интересно тем, что оно позволяет исследовать зависимость вероятности переобучения Q_ε от числа слоёв.

Теорема 11.4. *Пусть μ — пессимистичный метод минимизации эмпирического риска, $A = A_t$ — нижние t слоёв интервала булева куба с m пограничными и m_1 шумовыми объектами. Тогда для любого $\varepsilon \in [0, 1]$ вероятность переобучения есть*

$$Q_\varepsilon = \sum_{s=0}^{m} \sum_{s_1=0}^{m_1} \frac{C_m^s C_{m_1}^{s_1} C_{L-m-m_1}^{\ell-s-s_1}}{C_L^\ell} \big[s_1 \leqslant \tfrac{\ell}{L}(m_1 + \min\{t, m-s\} - \varepsilon k) \big].$$

Доказательство. Обозначим через X_0, X_1, S соответственно множества всех внутренних, шумовых и пограничных объектов; а через s_0, s_1, s соответственно — число внутренних, шумовых и пограничных объектов, попавших в обучающую выборку X.

Поскольку метод μ пессимистичный, он всегда будет выбирать из A алгоритм, который не ошибается на всех обучающих пограничных объектах, но ошибается на всех контрольных пограничных объектах. Поэтому

$$\nu(\mu X, X) = \frac{s_1}{\ell}; \qquad \nu(\mu X, \bar{X}) = \frac{(m_1 - s_1) + \min\{t, m-s\}}{k}.$$

Число разбиений $X \sqcup \bar{X}$, при которых $|X_0 \cap X| = s_0,\ |X_1 \cap X| = s_1,\ |S \cap X| = s$, равно $C_{m_0}^{s_0} C_{m_1}^{s_1} C_m^s$. Следовательно, вероятность переобучения представима в виде

$$Q_\varepsilon = \sum_{\substack{s_0=0 \\ s_0+s_1+s=\ell}}^{m_0} \sum_{s_1=0}^{m_1} \sum_{s=0}^{m} \frac{C_{m_0}^{s_0} C_{m_1}^{s_1} C_m^s}{C_L^\ell} \left[\frac{(m_1 - s_1) + \min\{t, m-s\}}{k} - \frac{s_1}{\ell} \geqslant \varepsilon \right].$$

Чтобы получить утверждение теоремы, достаточно воспользоваться соотношениями $m_0 + m_1 + m = L$ и $s_0 + s_1 + s = \ell$ и преобразовать неравенство в квадратных скобках к виду $s_1 \leqslant \frac{\ell}{L}(m_1 + \min\{t, m-s\} - \varepsilon k)$. ■

Интервал булева куба является уникальным примером семейства, для которого эффект расслоения практически не способствует снижению переобучения.

Следствие 11.4.1. *Если μ — пессимистичная минимизация эмпирического риска, A_m — интервал ранга m булева куба, то вероятность переобучения Q_ε совпадает с вероятностью равномерного отклонения:*

$$Q_\varepsilon = \widetilde{Q}_\varepsilon = \mathsf{P}\Big[\max_{a\in A}\delta(a,X) \geqslant \varepsilon\Big].$$

Доказательство. Пессимистичный метод μ всегда выбирает из A_m алгоритм, который не ошибается на всех обучающих пограничных объектах, но ошибается на всех контрольных пограничных объектах, что эквивалентно максимизации $\delta(a,X)$. ■

Вычислительный эксперимент. На рис. 11.6 представлены графики зависимости вероятности переобучения Q_ε от числа нижних слоёв $t = m_1,\ldots,m_1{+}m$. Три эксперимента отличались длиной генеральной выборки (200, 400, 1000), при этом сохранялись пропорции $\frac{m}{L} = 0.2$ и $\frac{m_1}{L} = 0.05$, то есть генеральная выборка всегда содержала 20% пограничных и 5% шумовых объектов. На графиках также показаны вклады слоёв в значение функционала Q_ε. Только нижние слои дают ненулевые вклады (каждый второй слой вообще не вносит вклад в Q_ε в силу отношения $\frac{\ell}{L} = \frac{1}{2}$, этим объясняется зубчатость графиков вкладов). Оказывается, что 20% пограничных объектов — это настолько мощный интервал, что при всех трёх значениях L вероятность переобучения быстро достигает значения 1. Вероятность переобучения близка к нулю только если брать самые нижние слои интервала (не более 2% от длины выборки).

Интерпретации и выводы. Приходится констатировать отрицательный, по сути, результат. Интервал булева куба с «разумными» на первый взгляд параметрами m_1, m оказывается слишком богатым семейством алгоритмов. Оценки вероятности переобучения Q_ε для интервала близки к нулю лишь при очень низких значениях m_1, m. Отсюда следуют два вывода.

Во-первых, хорошая обобщающая способность вряд ли возможна, если в выборке есть значительное количество пограничных объектов, на которых алгоритмы семейства допускают ошибки всеми возможными способами. Фактически, доля таких объектов добавляется к величине переобученности. Может возникнуть подозрение, что это происходит из-за пессимистичности МЭР, однако оценка для рандомизированного МЭР практически столь же плоха (упражнение 11.8).

Во-вторых, использовать интервал булева куба как модель реальных семейств вряд ли целесообразно. Гипотеза о существовании слоя пограничных объектов представляется достаточно разумной. Однако в реальных задачах алгоритмами семейства реализуются, по всей видимости, далеко не все способы допустить ошибки на пограничных объектах. Возможно, более адекватной была бы модель, в которой тем или иным способом вводится характеристика «степени граничности» объектов и оценивается распределение этой характеристики в выборке.

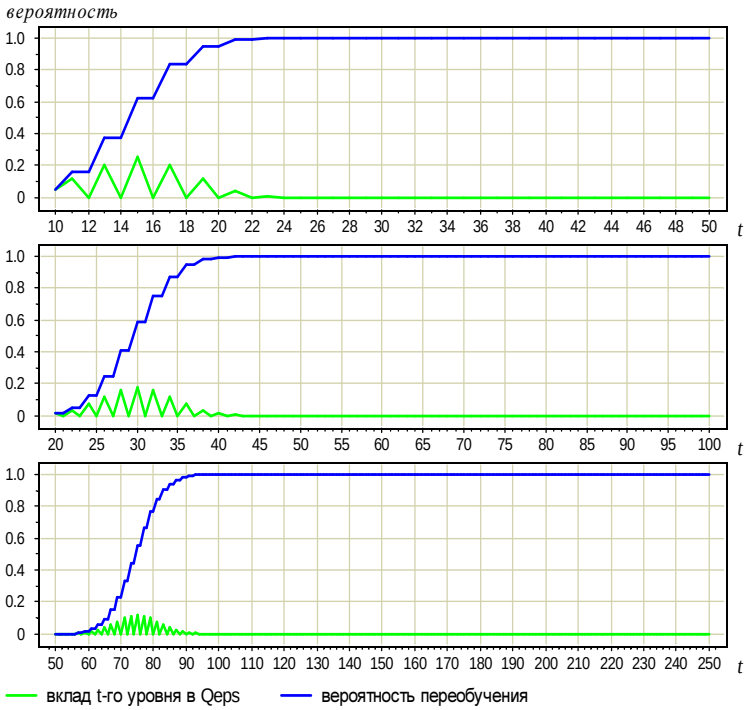


Рис. 11.6. Зависимость вероятности переобучения Q_ε от числа t нижних слоёв при $\varepsilon = 0.05$. Верхний график: $\ell = k = 100,\ m_1 = 10,\ m = 40$. Средний график: $\ell = k = 200,\ m_1 = 20,\ m = 80$. Нижний график: $\ell = k = 500,\ m_1 = 50,\ m = 200$.

§11.3 Блочная оценка

Допустим, что векторы ошибок всех алгоритмов $A = \{a_1, \ldots, a_D\}$ попарно различны, метод μ — пессимистичная минимизация эмпирического риска. Значения $I(a_d, x_i)$ образуют бинарную $L \times D$-матрицу ошибок, столбцы которой являются векторами ошибок алгоритмов, строки соответствуют объектам. Обозначим через $b = (b_1, \ldots, b_D)$ произвольный бинарный вектор размерности D. Выборка $\mathbb{X}$ разбивается на непересекающиеся *блоки* $U_b \subseteq \mathbb{X}$ так, что всем объектам в блоке соответствует одна и та же строка $b = (b_1, \ldots, b_D)$ в матрице ошибок:

$$U_b = \big\{x_i \in \mathbb{X} \;\big|\; I(a_d, x_i) = b_d,\ d = 1, \ldots, D\big\}.$$

Обозначим через B множество бинарных векторов b, которым соответствуют непустые блоки U_b. Очевидно, $|B| \leqslant \min\{L, 2^D\}$.

Обозначим $m_b = |U_b|$.

Каждой обучающей выборке $X \in [\mathbb{X}]^\ell$ поставим в соответствие целочисленный вектор $(s_b)_{b \in B}$ такой, что $s_b = |X \cap U_b|$ — число объектов из блока U_b, попадающих в обучающую выборку. Множество всех таких векторов, соответствующих всевозможным обучающим выборкам, обозначим через S. Очевидно, S можно также определить и другим способом:

$$S = \Big\{s = (s_b)_{b \in B} \;\Big|\; s_b = 0, \ldots, m_b,\ \sum_{b \in B} s_b = \ell\Big\}.$$

Запишем число ошибок алгоритма a_d на обучающей выборке X и контрольной выборке $\bar{X}$ в виде суммы по блокам:

$$n(a_d, X) = \sum_{b\in B} b_d\,|X \cap U_b| = \sum_{b\in B} b_d s_b;$$
$$n(a_d, \bar{X}) = \sum_{b\in B} b_d\,|\bar{X} \cap U_b| = \sum_{b\in B} b_d(m_b - s_b).$$

Таким образом, выбор алгоритма методом μ зависит только от того, сколько объектов s_b из каждого блока попадёт в обучающую выборку, но не зависит от того, какие именно это будут объекты. Определим функцию $d^*\colon S \to \{1,\ldots,D\}$ как номер алгоритма, выбранного методом μ по обучающей выборке. Если минимум $n(a,X)$ достигается на нескольких алгоритмах, то пессимистичный метод μ выбирает алгоритм с бóльшим $n(a,\bar{X})$. Если же и таких алгоритмов несколько, будем полагать, что выбирается алгоритм с бóльшим порядковым номером:

$$A(s) = \operatorname*{Arg\,min}_{d=1,\ldots,D} \sum_{b\in B} b_d s_b,$$
$$A'(s) = \operatorname*{Arg\,max}_{d\in A(s)} \sum_{b\in B} b_d(m_b - s_b),$$
$$d^*(s) = \max\bigl\{d\colon d \in A'(s)\bigr\}, \tag{11.2}$$

где через $\operatorname*{Arg\,min}_{d=1,\ldots,D} f(d)$ обозначается множество значений d, при которых функция $f(d)$ достигает минимального значения.

Теорема 11.5. *Пусть μ — пессимистичная минимизация эмпирического риска, векторы ошибок всех алгоритмов $a \in A$ попарно различны. Тогда вероятность получить алгоритм a_d в результате обучения:*

$$\mathsf{P}\bigl[\mu X{=}a_d\bigr] = \frac{1}{C_L^\ell} \sum_{s\in S} \biggl(\prod_{b\in B} C_{m_b}^{s_b}\biggr) \bigl[\,d^*(s){=}d\,\bigr]; \tag{11.3}$$

вероятность переобучения:

$$Q_\varepsilon = \frac{1}{C_L^\ell} \sum_{s\in S} \biggl(\prod_{b\in B} C_{m_b}^{s_b}\biggr) \biggl[\sum_{b\in B} b_{d^*(s)}\bigl(m_b\ell - s_b L\bigr) \geqslant \varepsilon k\ell\biggr]. \tag{11.4}$$

Доказательство.

Произвольному набору значений $(s_b)_{b\in B}$ из S соответствует множество выборок $X \in [\mathbb{X}]^\ell$ таких, что $|X \cap U_b| = s_b$. Число таких выборок равно произведению $\prod\limits_{b\in B} C_{m_b}^{s_b}$, так как для каждого блока U_b существует $C_{m_b}^{s_b}$ способов отобрать s_b объектов в подвыборку $X \cap U_b$.

Поскольку условия $\mu X = a_d$ и $d^*(s) = d$ равносильны, вероятность получить алгоритм a_d в результате обучения выражается в следующем виде:

$$\mathsf{P}\bigl[\mu X{=}a_d\bigr] = \frac{1}{C_L^\ell} \sum_{X\in[\mathbb{X}]^\ell} \bigl[d^*(s){=}d\bigr] = \frac{1}{C_L^\ell} \sum_{s\in S} \biggl(\prod_{b\in B} C_{m_b}^{s_b}\biggr) \bigl[\,d^*(s){=}d\,\bigr].$$

Теперь запишем вероятность переобучения:

$$Q_\varepsilon = \mathsf{P}\big[\delta_\mu(X) \geqslant \varepsilon\big] = \mathsf{P}\sum_{d=1}^{D}\big[\mu X{=}a_d\big]\big[\delta(a_d, X) \geqslant \varepsilon\big].$$

Распишем отклонение частот ошибок алгоритма a_d в виде суммы по блокам:

$$\delta(a_d, X) = \frac{1}{k}\sum_{b\in B} b_d(m_b - s_b) - \frac{1}{\ell}\sum_{b\in B} b_d s_b = \frac{1}{\ell k}\sum_{b\in B} b_d\big(m_b\ell - s_b L\big).$$

Тогда выражение для вероятности переобучения примет вид:

$$Q_\varepsilon = \frac{1}{C_L^\ell}\sum_{s\in S}\Big(\prod_{b\in B} C_{m_b}^{s_b}\Big)\sum_{d=1}^{D}\big[\,d^*(s){=}d\,\big]\left[\sum_{b\in B} b_d\big(m_b\ell - s_b L\big) \geqslant \varepsilon\ell k\right].$$

Отсюда немедленно вытекает требуемое выражение (11.4). ■

Объём вычислений по формулам (11.3) и (11.4) экспоненциален по длине выборки L. В худшем случае, когда все блоки U_b одноэлементные, множество S состоит из всевозможных булевых векторов длины L, содержащих ровно ℓ единиц. Тогда число слагаемых в (11.3) и (11.4) равно C_L^ℓ.

Вычисления по Теореме 11.5 эффективны только когда число блоков $|B|$ невелико, в частности, при малом числе алгоритмов. Ещё один недостаток блочной оценки в том, что она в явном виде не учитывает свойства расслоения и связности.

§11.4 Пара алгоритмов

Рассмотрим семейство из двух алгоритмов $A = \{a_1, a_2\}$. Чтобы воспользоваться блочной оценкой, положим $B = (11, 10, 01, 00)$.

Пусть в выборке $\mathbb{X}$ имеется m_{11} объектов, на которых оба алгоритма допускают ошибку; m_{10} объектов, на которых только a_1 допускает ошибку; m_{01} объектов, на которых только a_2 допускает ошибку; $m_{00} = L - m_{11} - m_{10} - m_{01}$ объектов, на которых оба алгоритма дают верный ответ:

$$\vec{a}_1 = (\,1,\ldots,1,\ 1,\ldots,1,\ 0,\ldots,0,\ 0,\ldots,0\,);$$
$$\vec{a}_2 = (\underbrace{1,\ldots,1}_{m_{11}},\ \underbrace{0,\ldots,0}_{m_{10}},\ \underbrace{1,\ldots,1}_{m_{01}},\ \underbrace{0,\ldots,0}_{m_{00}}).$$

Теорема 11.6. *Пусть μ — пессимистичная минимизация эмпирического риска, и семейство состоит из двух алгоритмов $A = \{a_1, a_2\}$. Тогда для любого $\varepsilon \in [0, 1)$*

$$Q_\varepsilon = \sum_{s_{11}=0}^{m_{11}}\sum_{s_{10}=0}^{m_{10}}\sum_{s_{01}=0}^{m_{01}} \frac{C_{m_{11}}^{s_{11}} C_{m_{10}}^{s_{10}} C_{m_{01}}^{s_{01}} C_{L-m_{11}-m_{10}-m_{01}}^{\ell-s_{11}-s_{10}-s_{01}}}{C_L^\ell} \times$$
$$\times\Big(\big[s_{10} < s_{01}\big]\big[s_{11} + s_{10} \leqslant \tfrac{\ell}{L}(m_{11} + m_{10} - \varepsilon k)\big] +$$
$$+\,\big[s_{10} \geqslant s_{01}\big]\big[s_{11} + s_{01} \leqslant \tfrac{\ell}{L}(m_{11} + m_{01} - \varepsilon k)\big]\Big). \qquad (11.5)$$

Доказательство. Воспользуемся Теоремой 11.5. Множество S состоит из целочисленных векторов $s = (s_{11}, s_{10}, s_{01}, s_{00})$, для которых $s_{11} + s_{10} + s_{01} + s_{00} = \ell$. Поэтому сумма $\sum_{s \in S}$ преобразуется в тройную сумму $\sum_{s_{11}=0}^{m_{11}} \sum_{s_{10}=0}^{m_{10}} \sum_{s_{01}=0}^{m_{01}}$, при этом s_{00} выражается через остальные компоненты вектора s. Номер $d^*(s)$ алгоритма, выбранного методом μ по обучающей выборке, равен 1 при $s_{10} < s_{01}$ и 2 при $s_{10} \geqslant s_{01}$. Теперь подставим значения m_b, s_b, $d^*(s)$ в (11.4):

$$\left[\sum_{b \in B} b_{d^*(s)}\big(m_b\ell - s_b L\big) \geqslant \varepsilon\ell k\right] = \big[d^*(s)=1\big]\big[(m_{10}+m_{11})\ell - (s_{10}+s_{11})L \geqslant \varepsilon\ell k\big] + \\ + \big[d^*(s)=2\big]\big[(m_{01}+m_{11})\ell - (s_{01}+s_{11})L \geqslant \varepsilon\ell k\big].$$

Отсюда следует требуемая оценка (11.5). ■

Вычислительный эксперимент. На Рис. 11.7 и 11.8 показаны зависимости вероятности переобучения Q_ε и оценки ЭЛКР $\hat{\Delta}$ от различности алгоритмов $\rho(a_1, a_2) = m_{01} + m_{10}$ при $\ell = k = 100$, $\varepsilon = 0.05$. Тонкими линиями показаны оценки ЭЛКР, вычисленные *методом Монте-Карло* по 1000 случайных разбиений.

Графики позволяют сделать следующие выводы.

1. VC-оценка ЭЛКР $\hat{\Delta} = 2$ достигается лишь в том случае, когда нет расслоения (алгоритмы допускают на $\mathbb{X}$ одинаковое число ошибок, $m_{01} = m_{10}$) и нет сходства (алгоритмы максимально различны, $m_{00} = m_{11} = 0$).

2. Чем сильнее расслоение ($d = m_{01} - m_{10} > 0$), тем меньше вероятность переобучения, и тем ближе ЭЛКР к 1. В данном эксперименте при $d = 40$ худший из двух алгоритмов уже практически никогда не выбирается.

3. Чем сильнее сходство ($m_{01}, m_{10} \to 0$), тем меньше вероятность переобучения и тем ближе ЭЛКР к 1. С точки зрения переобучения два схожих алгоритма ведут себя практически как один алгоритм.

Основной вывод. Даже в простейшем случае, когда алгоритмов только два, уже возникает явление переобучения и проявляются эффекты расслоения и сходства, снижающие вероятность переобучения.

Резюме

Точные оценки вероятности переобучения к данному моменту известны для ряда модельных семейств алгоритмов: монотонных и унимодальных цепей и сетей, интервалов булева куба и их нижних слоёв, и некоторых других. Эти оценки полезны для понимания того, как свойства расслоения и связности влияют на вероятность переобучения. Кроме того, модельные семейства играют роль испытательного полигона при создании комбинаторной техники оценивания переобучения. Однако для непосредственного практического применения модельные семейства не годятся.

В следующей лекции мы рассмотрим теоретико-групповой подход, который позволяет выводить точные оценки вероятности переобучения для модельных семейств, обладающих свойством симметрии.

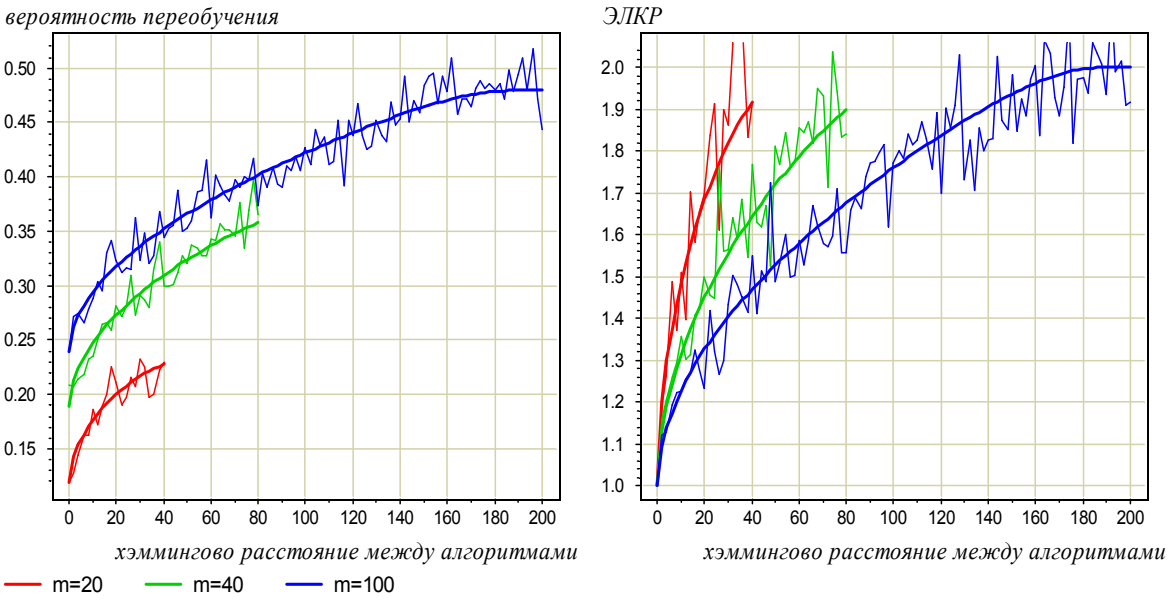


Рис. 11.7. Зависимость вероятности переобучения Q_ε и оценки ЭЛКР от различности алгоритмов, когда они допускают одинаковое число ошибок, $m_{01} = m_{10}$. Три графика соответствуют трём значениям числа ошибок на полной выборке $m = m_{01} + m_{11} \in \{20, 40, 100\}$. Тонкими линиями показаны эмпирические оценки методом Монте-Карло по 1000 случайных разбиений.

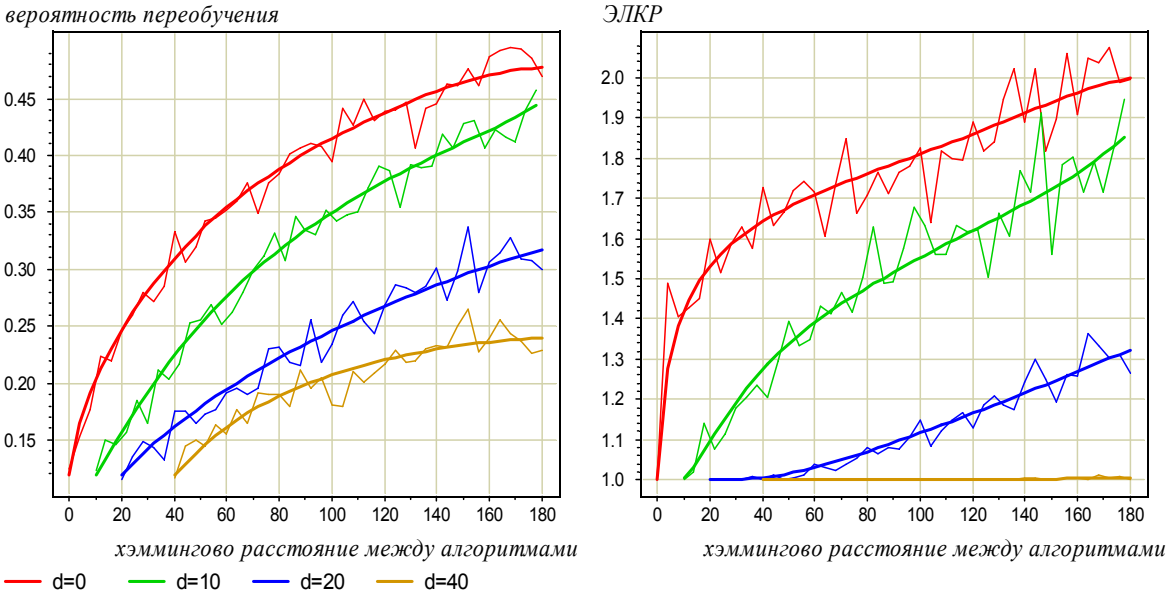


Рис. 11.8. Зависимость вероятности переобучения и оценки ЭЛКР от различности алгоритмов, когда $m_{11} = 20$ и второй алгоритм допускает на d ошибок больше, $m_{01} = m_{10} + d$. Четыре графика соответствуют четырём разным значениям $d \in \{0, 10, 20, 40\}$. Тонкими линиями показаны эмпирические оценки методом Монте-Карло по 1000 случайных разбиений.

Упражнения

Задача 11.1 (1). Привести пример выборки в $\mathbb{R}^2$, для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

Задача 11.2 (3*). Привести пример выборки в $\mathbb{R}^h$, для которой множество линейных классификаторов (не обязательно всех возможных) порождает монотонную сеть алгоритмов.

Задача 11.3 (5*). Сформулировать требования к семейству алгоритмов, при которых оценка расслоения–связности является точной.

Задача 11.4 (5). Получить оценку вероятности переобучения для случая, когда по каждой размерности d задана своя высота H_d.

Задача 11.5 (5). Получить оценку вероятности переобучения для унимодальной h-мерной сетки.

Задача 11.6 (5). Получить оценку вероятности переобучения для пучка h монотонных цепей длины H.

Задача 11.7 (5*). Получить оценку вероятности переобучения для t нижних слоёв интервала булева куба через порождающие и запрещающие множества.

Задача 11.8 (2). Получить оценку вероятности переобучения для t нижних слоёв интервала булева куба в случае рандомизированного метода МЭР.

Задача 11.9 (1). Верно ли Следствие 11.4.1 для семейства A_t, $t < m$?

Задача 11.10 (1). Получить оценку вероятности переобучения для пары алгоритмов, пользуясь принципом порождающих и запрещающих множеств.

Список литературы

[1] *Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д.* Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.

[2] *Алимов Ю. И.* Альтернатива методу математической статистики. — Знание, 1980.

[3] *Беляев Ю. К.* Вероятностные методы выборочного контроля. — М.: Наука, 1975.

[4] *Большев Л. Н., Смирнов Н. В.* Таблицы математической статистики. — М.: Наука, 1983.

[5] *Ботов П. В.* Точные оценки вероятности переобучения для монотонных и унимодальных семейств алгоритмов // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 7–10.
http://www.mmro.ru.

[6] *Вапник В. Н.* Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.

[7] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *ДАН СССР.* — 1968. — Т. 181, № 4. — С. 781–784.

[8] *Вапник В. Н., Червоненкис А. Я.* О равномерной сходимости частот появления событий к их вероятностям // *Теория вероятностей и ее применения.* — 1971. — Т. 16, № 2. — С. 264–280.

[9] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[10] *Воронцов К. В.* Обзор современных исследований по проблеме качества обучения алгоритмов // *Таврический вестник информатики и математики.* — 2004. — № 1. — С. 5–24.
http://www.ccas.ru/frc/papers/voron04twim.pdf.

[11] *Воронцов К. В., Решетняк И. М.* Точные комбинаторные оценки обобщающей способности онлайнового обучения // Межд. конф. Интеллектуализация обработки информации ИОИ-8. — М.: МАКС Пресс, 2010. — С. 24–27.
http://iip.mmro.ru.

[12] *Гаек Я., Шидак З.* Теория ранговых критериев. — М.: Наука, 1971.

[13] *Гоппа В. Д.* Введение в алгебраическую теорию информации. — М.: Наука, 1995.

[14] *Грэхем Р., Кнут Д., Паташник О.* Конкретная математика. — М.: Мир, 1998.

[15] *Гуров С. И.* Точечная оценка вероятности 0-события // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 22–25.
http://www.mmro.ru.

[16] *Донской В. И.* Колмогоровская сложность классов общерекурсивных функций с ограниченной ёмкостью // *Таврический вестник информатики и математики.* — 2005. — № 1. — С. 25–34.
http://www.ccas.ru/frc/papers/donskoy05kolmogorov.pdf.

[17] *Дюличева Ю. Ю.* Оценка VCD r-редуцированного эмпирического леса // *Таврический вестник информатики и математики.* — 2003. — № 1. — С. 31–42.
http://www.ccas.ru/frc/papers/dulicheva03vcdforest.pdf.

[18] *Журавлёв Ю. И.* Об алгебраическом подходе к решению задач распознавания или классификации // *Проблемы кибернетики.* — 1978. — Т. 33. — С. 5–68.

http://www.ccas.ru/frc/papers/zhuravlev78prob33.pdf.

[19] *Журавлёв Ю. И., Рязанов В. В., Сенько О. В.* «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006.

[20] *Загоруйко Н. Г.* Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999.

[21] *Загоруйко Н. Г., Ёлкина В. Н., Лбов Г. С.* Алгоритмы обнаружения эмпирических закономерностей. — Новосибирск: Наука, 1985.

[22] *Колмогоров А. Н.* Теория информации и теория алгоритмов / Под ред. Ю. В. Прохоров. — М.: Наука, 1987. — 304 с.

[23] *Кочедыков Д. А.* Структуры сходства в семействах алгоритмов классификации и оценки обобщающей способности // Всеросс. конф. Математические методы распознавания образов-14. — М.: МАКС Пресс, 2009. — С. 45–48.
http://www.mmro.ru.

[24] *Матросов В. Л.* Корректные алгебры ограниченной ёмкости над множествами некорректных алгоритмов // *ДАН СССР*. — 1980. — Т. 253, № 1. — С. 25–30.
http://www.ccas.ru/frc/papers/matrosov80dan.pdf.

[25] *Матросов В. Л.* Ёмкость алгебраических расширений модели алгоритмов вычисления оценок // *ЖВМиМФ*. — 1984. — Т. 24, № 11. — С. 1719–1730.

[26] *Матросов В. Л.* Нижние границы ёмкости многомерных алгебр алгоритмов вычисления оценок // *ЖВМиМФ*. — 1984. — Т. 24, № 12. — С. 1881–1892.

[27] *Матросов В. Л.* Ёмкость алгоритмических многочленов над множеством алгоритмов вычисления оценок // *ЖВМиМФ*. — 1985. — Т. 25, № 1. — С. 122–133.

[28] *Райгородский А. М.* Экстремальные задачи теории графов и анализ данных. — М.: НИЦ «Регулярная и хаотическая динамика», Институт компьютерных исследований, 2008. — 118 с.

[29] *Смирнов Н. В.* Оценка расхождения между эмпирическими кривыми распределения в двух независимых выборках // *Бюлл. Московского ун-та, серия А*. — 1939. — № 2. — С. 3–14.

[30] *Anthony M., Shawe-Taylor J.* A result of Vapnik with applications // *Discrete Applied Mathematics*. — 1993. — Vol. 47, no. 2. — Pp. 207–217.
http://citeseer.ist.psu.edu/anthony91result.html.

[31] *Bartlett P.* Lower bounds on the Vapnik-Chervonenkis dimension of multi-layer threshold networks // Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory. — ACM Press, New York, NY, 1993. — Pp. 144–150.
http://citeseer.ist.psu.edu/bartlett93lower.html.

[32] *Bauer M., Godreche C., Luck J. M.* Statistics of persistent events in the binomial random walk: Will the drunken sailor hit the sober man? // *J.STAT.PHYS*. — 1999. — Vol. 96. — P. 963.
http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/9905252.

[33] *Bax E. T.* Similar classifiers and VC error bounds: Tech. Rep. CalTech-CS-TR97-14: 1997.
http://citeseer.ist.psu.edu/bax97similar.html.

[34] *Boucheron S., Bousquet O., Lugosi G.* Theory of classification: A survey of some recent advances // *ESAIM: Probability and Statistics*. — 2005. — no. 9. — Pp. 323–375.

http://www.econ.upf.edu/∼lugosi/esaimsurvey.pdf.

[35] *Breiman L.* Technical note: Some properties of splitting criteria // *Machine Learning.* — 1996. — Vol. 24. — Pp. 41–47.
http://http://www.cba.ua.edu/~mhardin/BreimanMachineLearning1996.pdf.

[36] *Chvátal V.* The tail of the hypergeometric distribution // *Discrete Mathematics.* — 1979. — Vol. 25, no. 3. — Pp. 285–287.

[37] *Cohen W. W., Singer Y.* A simple, fast and effective rule learner // Proc. of the 16 National Conference on Artificial Intelligence. — 1999. — Pp. 335–342.
http://citeseer.ist.psu.edu/198445.html.

[38] *Freund Y., Schapire R. E.* Experiments with a new boosting algorithm // International Conference on Machine Learning. — 1996. — Pp. 148–156.
http://citeseer.ist.psu.edu/freund96experiments.html.

[39] *Hastie T., Tibshirani R., Friedman J.* The Elements of Statistical Learning, 2nd ed. — Springer, 2009.

[40] *Herbrich R., Williamson R. C.* Learning and generalization: theoretical bounds. — Cambridge, MA, USA: MIT Press, 2002. — Pp. 619–623.

[41] *Karpinski M., Macintyre A.* Polynomial bounds for VC dimension of sigmoidal neural networks // 27th ACM Symposium on Theory of Computing, Las Vegas, Nevada, US. — 1995. — Pp. 200–208.
http://citeseer.ist.psu.edu/karpinski95polynomial.html.

[42] *Kearns M. J., Mansour Y., Ng A. Y., Ron D.* An experimental and theoretical comparison of model selection methods // 8th Conf. on Computational Learning Theory, Santa Cruz, California, US. — 1995. — Pp. 21–30.
http://citeseer.ist.psu.edu/kearns95experimental.html.

[43] *Langford J.* Quantitatively Tight Sample Complexity Bounds: Ph.D. thesis / Carnegie Mellon Thesis. — 2002.
http://citeseer.ist.psu.edu/langford02quantitatively.html.

[44] *Langford J., McAllester D.* Computable shell decomposition bounds // Proc. 13th Annu. Conference on Comput. Learning Theory. — Morgan Kaufmann, San Francisco, 2000. — Pp. 25–34.
http://citeseer.ist.psu.edu/langford00computable.html.

[45] *Lugosi G.* On concentration-of-measure inequalities. — Machine Learning Summer School, Australian National University, Canberra. — 2003.
http://citeseer.ist.psu.edu/lugosi98concentrationmeasure.html.

[46] *Martin J. K.* An exact probability metric for decision tree splitting and stopping // *Machine Learning.* — 1997. — Vol. 28, no. 2-3. — Pp. 257–291.
http://citeseer.ist.psu.edu/martin97exact.html.

[47] *Mazurov V., Khachai M., Rybin A.* Committee constructions for solving problems of selection, diagnostics and prediction // *Proceedings of the Steklov Institute of mathematics.* — 2002. — Vol. 1. — Pp. 67–101.
http://tom.imm.uran.ru/khachay/publications/mine/psis67.pdf.

[48] *Mertens S., Engel A.* Vapnik-Chervonenkis dimension of neural networks with binary weights // *Phys. Rev. E.* — 1997. — Vol. 55, no. 4. — Pp. 4478–4488.

[49] *Mullin M., Sukthankar R.* Complete cross-validation for nearest neighbor classifiers // Proceedings of International Conference on Machine Learning. — 2000. — Pp. 639–646. http://citeseer.ist.psu.edu/309025.html.

[50] *Quinlan J.* Induction of decision trees // *Machine Learning.* — 1986. — Vol. 1, no. 1. — Pp. 81–106.

[51] *Rissanen J.* Modeling by shortest data description // *Automatica.* — 1978. — Vol. 14. — Pp. 465–471.

[52] *Schapire R. E., Singer Y.* Improved boosting using confidence-rated predictions // *Machine Learning.* — 1999. — Vol. 37, no. 3. — Pp. 297–336. http://citeseer.ist.psu.edu/article/singer99improved.html.

[53] *Sill J.* Monotonicity and connectedness in learning systems: Ph.D. thesis / California Institute of Technology. — 1998. http://etd.caltech.edu/etd/available/etd-09222005-110351/.

[54] *Valiant L. G.* A theory of the learnable // *Communications of the ACM.* — 1984. — Vol. 27. — Pp. 1134–1142.

[55] *Vapnik V.* Estimation of Dependencies Based on Empirical Data. — Springer-Verlag, New York, 1982.

[56] *Vapnik V.* The nature of statistical learning theory. — Springer-Verlag, New York, 1995.

[57] *Vapnik V.* Statistical Learning Theory. — Wiley, New York, 1998.

[58] *Vayatis N., Azencott R.* Distribution-dependent Vapnik-Chervonenkis bounds // *Lecture Notes in Computer Science.* — 1999. — Vol. 1572. — Pp. 230–240. http://citeseer.ist.psu.edu/vayatis99distributiondependent.html.

[59] *Vorontsov K. V.* Combinatorial probability and the tightness of generalization bounds // *Pattern Recognition and Image Analysis.* — 2008. — Vol. 18, no. 2. — Pp. 243–259. http://www.springerlink.com/content/78537p01838123u7/.

[60] *Vorontsov K. V.* On the influence of similarity of classifiers on the probability of overfitting // Pattern Recognition and Image Analysis: new information technologies (PRIA-9). — Vol. 2. — Nizhni Novgorod, Russian Federation, 2008. — Pp. 303–306. http://www.ccas.ru/frc/papers/voron08pria-conf-eng.pdf.

[61] *Vorontsov K. V.* Splitting and similarity phenomena in the sets of classifiers and their effect on the probability of overfitting // *Pattern Recognition and Image Analysis.* — 2009. — Vol. 19, no. 3. — Pp. 412–420. http://www.MachineLearning.ru/wiki/images/0/0e/Voron09roai2008.pdf.

[62] *Vorontsov K. V.* Exact combinatorial bounds on the probability of overfitting for empirical risk minimization // *Pattern Recognition and Image Analysis.* — 2010. — Vol. 20, no. 3. — Pp. 269–285.